2-2013

Accurate and Robust Mechanical Modeling of Proteins

Naomi Fox
University of Massachusetts Amherst, fox@cs.umass.edu

ACCURATE AND ROBUST MECHANICAL MODELING OF PROTEINS

A Dissertation Presented

by

NAOMI K. FOX

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2013

Computer Science

ACCURATE AND ROBUST MECHANICAL MODELING OF PROTEINS

A Dissertation Presented

by

NAOMI K. FOX

Approved as to style and content by:

Ileana Streinu, Chair

Lori A. Clarke, Member

Jeanne Hardy, Member

Ion Mandoiu, Member

Jack Wileden, Member

Lori A. Clarke, Department Chair
Computer Science

ACKNOWLEDGMENTS

I am so grateful for all of the support and opportunities that have helped me to achieve this dream of attaining a Ph.D. I would first like to thank my advisor, Ileana Streinu, for her years of guidance. She brought me into research when I was an undergrad at Smith and continued to foster my interest in research through grad school. She has taught me how far optimism and sheer perseverance can take you in research. I am also grateful to the rest of my dissertation committee, Lori Clarke, Jeanne Hardy, Ion Mandoiu, and Jack Wileden, for providing thoughtful and helpful comments throughout the process.

I would like to acknowledge my funding sources, mainly Ileana's NFS and DARPA grants. I was also privileged to receive travel funding to attend a number of workshops and conferences. In particular I am thankful for the interdisciplinary research experiences that I gained at the intense Barbados Workshops.

Mentoring and peer support have made a key difference for me. The CS women's events were an important outlet during my days at the UMass Computer Science building. My experiences at three CRA-W Cohort Workshops and at the Tapia Conference and Doctoral Consortium were very enriching. I feel lucky to have had such wonderful peers in LinKaGe: Anastasia, Ashraf, Audrey, John, Louis, Yang, and especially Filip, my partner-in-crime in building KINARI.

I would like to thank my family for their boundless encouragement, Tiffiniy for being such a bright star in my life, and my friends Allegra, Ashmita, Becca, Philipp, and Stephanie for their support. And finally, I am thankful to Greg, my future husband, for all the happiness he brings into my life.

ABSTRACT

ACCURATE AND ROBUST MECHANICAL MODELING OF PROTEINS

FEBRUARY 2013

NAOMI K. FOX

B.A., SMITH COLLEGE

B.E., DARTMOUTH COLLEGE

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Ileana Streinu

Through their motion, proteins perform essential functions in the living cell. Although we cannot observe protein motion directly, over 68,000 crystal structures are freely available from the Protein Data Bank. Computational protein rigidity analysis systems leverage this data, building a mechanical model from atoms and pairwise interactions determined from a static 3D structure. The rigid and flexible components of the model are then calculated with a pebble game algorithm, predicting a protein's flexibility with much more computational efficiency than physical simulation. In prior work with rigidity analysis systems, the available modeling options were hard-coded, and evaluation was limited to case studies.

The focus of this thesis is improving accuracy and robustness of rigidity analysis systems. The first contribution is in new approaches to mechanical modeling of non-covalent interactions, namely hydrogen bonds and hydrophobic interactions. Unlike

covalent bonds, the behavior of these interactions varies with their energies. I systematically investigate energy-refined modeling of these interactions. Included in this is a method to assign a score to a predicted cluster decomposition, adapted from the B-cubed score from information retrieval. Another contribution of this thesis is in new approaches to measuring the robustness of rigidity analysis results. A protein's fold is held in place by weak noncovalent interactions, known to break and form during natural fluctuations. Rigidity analysis has been conventionally performed on only a single snapshot, rather than on an entire trajectory, and no information was made available on the sensitivity of the clusters to variations in the interaction network. I propose an approach to measure the robustness of rigidity results, by studying how detrimental the loss of a single interaction may be to a cluster's rigidity. The accompanying study shows that, when present, highly critical interactions are concentrated around the active site, indicating that nature has designed a very versatile system for transitioning between unique conformations.

Over the course of this thesis, we develop the KINARI library for experimenting with extensions to rigidity analysis. The modular design of the software allows for easy extensions and tool development. A specific feature is the inclusion of several modeling options, allowing more freedom in exploring biological hypotheses and future benchmarking experiments.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Through their motion, proteins perform essential functions in the living cell. Their mechanical and functional properties are dependent on their 3D shapes. In recent years, the growth of the Protein Data Bank has provided a wealth of structural data, with over 68,000 protein crystal structure data files freely available to download. For many well-studied proteins, structural data on multiple unique conformations are often available [18]. However, typically these structures are limited to the low energy conformations. Available laboratory methods to study conformational change do not produce atom-level data.

Computational methods hold promise in inferring protein motion from laboratory experimental data. The most accurate and extensively-developed of these is physics-based simulation, or molecular dynamics (MD). Conventional MD operates on a very fine-grained representation of a macromolecule, incorporating all atoms and relevant pairwise inter-atomic forces. Although its accuracy for studying atomic-level protein motions is unmatched [39], MD proves to be quite computationally expensive. For example, the first millisecond timescale MD simulation on a 58 residue protein required over two months to compute on the Anton supercomputer in 2009 [57].

An alternative computational approach, protein rigidity analysis processes molecular data and provides a course-grained representation describing the mechanics of the molecule. The course-grained representation can be studied directly, or leveraged in motion generation methods [34]. Figure 1.1 shows the steps undertaken by our protein rigidity analysis system, KINARI. Using the set of atoms in the PDB

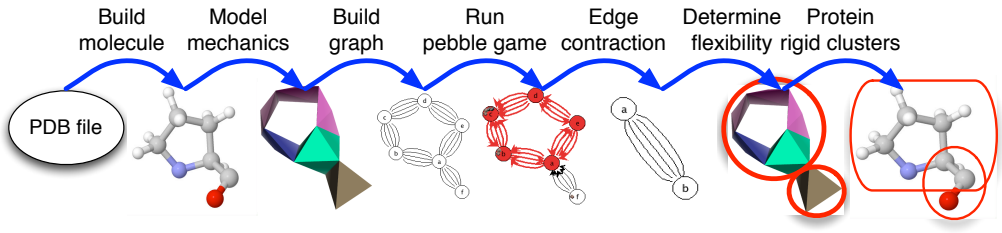


Figure 1.1: Steps of Protein Rigidity Analysis

and calculated inter-atomic interactions, a mechanical model of the protein is built. The rigid and flexible regions of the model are then determined, with mathematical guarantees, via an efficient graph-based algorithm. For a typical 100-residue protein, rigidity analysis is completed in seconds. Because rigidity analysis is so fast, it is feasible to analyze very large data sets.

Accuracy in the prediction of the flexible and rigid regions of the protein is of key importance for understanding the protein's behavior. The inclusion or exclusion of one inter-atomic interaction in the input can have a drastic impact on the rigidity results. In proteins, the function of the active site, where binding with other molecules occurs, is often dictated by the mobility of adjacent loops. A method with strong predictive power would successfully capture the flexibility of the loops within the small region, while still detecting domains in which the relative mobility between atoms is low.

Protein rigidity analysis was pioneered by Jacobs, Thorpe, and collaborators, and implemented in the software MSU-FIRST, ASU-FIRST, and the Flexweb server [10, 45, 48] (`http://flexweb.asu.edu`). In case studies on four proteins, flexibility predictions by the MSU-FIRST software were shown to correlate well with experimental evidence [47, 48]. But this validation was insufficient to prove that the same performance could be expected on any protein. Also, many design decisions were hard-coded into the software, hindering the development and testing of new modeling methods.

1.1 Thesis contributions

We now describe the three main contributions made in this thesis. For the first contribution, we have developed KINARI, an experimental software platform for validating the predictive power of protein rigidity analysis. This was a collaborative effort within Streinu's Linkage lab, to address the need to experiment with different modeling options. The second and third contributions are refinements we have made towards this validation effort. The first refinement is toward improving modeling accuracy of noncovalent interactions with new approaches. As part of this contribution, we discuss the development of an evaluation and benchmarking methodology. The second refinement is toward characterizing the robustness of rigidity analysis results.

In the next three subsections, we provide the motivations for each of these contributions, as well as a short summary of the work we have undertaken.

1.1.1 KINARI: software for rigidity analysis, with applications in molecular modeling

In prior rigidity analysis systems, modeling and curation options were limited, and the software was not designed for modeling experimentation. In particular, the mechanical model representation in ASU-FIRST was incorrect, resulting in rigid clusters which were not maximal. Also, no software libraries were available for performing mechanical modeling or rigidity analysis. To address these limitations, we have have developed KINARI to provide a general, well-tested, versatile library for rigidity analysis of molecular structures (not just proteins), which is easy to integrate in larger applications.

In this thesis, we describe the concepts behind KINARI. We also give an overview of the software architecture and detail how the modeling features are implemented. Figure 1.2 shows the main components that fall under the KINARI project. All work was completed under the direction of Prof. Ileana Streinu, who designed the soft-

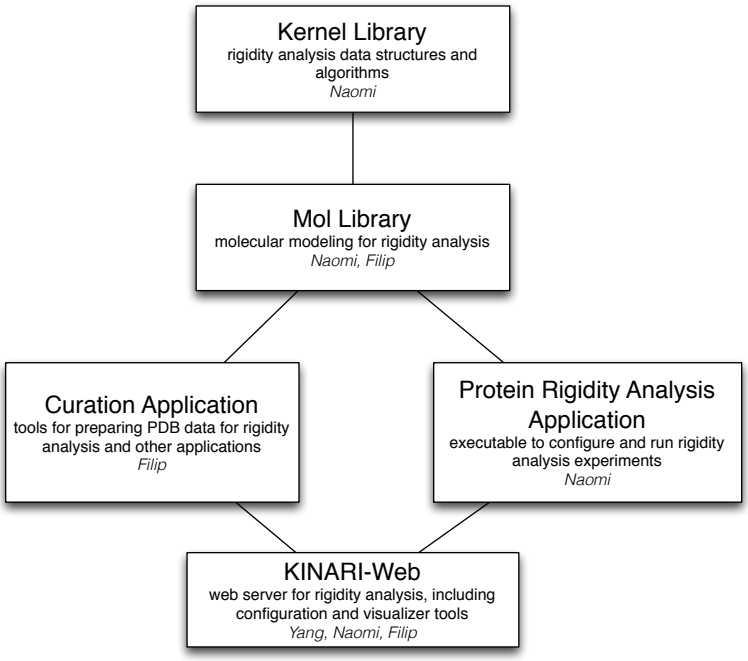


Figure 1.2: KINARI Components. The names of the software engineer contributors are listed for each component. The core technology of our rigidity analysis applications lies in the Kernel library.

ware and contributed the mechanical modeling scheme. As part of my thesis work, I implemented the kernel library and modeling code in the molecular library. Filip Jagodzinski implemented the non-trivial task of PDB file parsing, and performed much of the profiling. His thesis work focuses on correlating rigidity metrics with protein stability data derived from laboratory experiments. Yang Li, Smith 2011, wrote the bulk of the code for the web front-end and Jmol-based visualizer. We released the KINARI-Web server in 2011 with a publication in the NAR special web server issue [21]. We followed this release with a tutorial at BIBM 2011 [80]. The kernel library, was released under the name KINARI-Lib, with a corresponding tutorial, in June of 2012 at the Minisymposium on Publicly Available Geometric/Topological Software [22].

1.1.2 Toward improving modeling accuracy in protein rigidity analysis systems

Building on KINARI, we propose new approaches to mechanical modeling of non-covalent interactions, namely hydrogen bonds and hydrophobic interactions. These interactions are mainly responsible for stabilizing a protein's 3D fold. In previous work, hydrogen bonds were modeled as mechanically equivalent to covalent bonds, fixing bond length and bond angles at incident atoms [10, 21, 32, 48]. It has been observed early on that such a method may lead to inaccurate results, such as an almost complete rigidification of the protein model. Since it is known that not all hydrogen bonds have the same strength, an energy function was applied to prune the weakest bonds and exclude them from the model [48]. A universal hydrogen bond energy cutoff, which would produce biologically credible results for any protein input, has never been found. Wells *et al.* have pointed out the discrepancies in the chosen hydrogen bond energy cutoffs used in a number of previous studies in the literature [94].

Also in previous systems, hydrophobic interactions were identified with heuristic approaches [10], and, unlike hydrogen bonds, they had no associated energies. It has been observed that the tuning of the hydrophobic interactions can be just as important as for hydrogen bonds. Gohlke *et al.* [32] comment, in their study of flexibility changes during Ras-Raf complex formation, "Finding the appropriate balance between these interactions [hydrogen bonds and hydrophobics] is thus crucial for an accurate representation of the flexibility characteristics of proteins". Thus, we pose the following question.

Can an energy-differentiated modeling scheme improve accuracy of protein rigidity analysis?

A limitation of prior work is insufficient evaluation. Although MSU-FIRST proved to have non-trivial performance at predicting rigid and flexible regions on a number of case studies, it was not clear that this performance would generalize to other proteins.

Indeed, in a number of studies which followed using the software, it was reported that a fair amount of tuning was required [32,94]. The convention established for the MSU-FIRST and ASU-FIRST software was to tune with the hydrogen bond energy cutoff. There were no guidelines on how to do this, nor was there a quantitative way to measure the accuracy of these systems.

Contribution. We propose two new methods for incorporating noncovalent interactions for protein rigidity analysis. First, rather than simply removing weaker hydrogen bonds, we propose varying the way that the hydrogen bonds are modeled, based on their strength. We investigate modeling the weak hydrogen bonds as a rigid bar which fixes the distance between the endpoints, but permits full rotational freedom. We reveal the limitations of the current mathematical theory for supporting this modeling, and propose heuristics to approximate the rigidity results. The second method we propose is in the inclusion of hydrophobic interactions. Rather than using a heuristic as first proposed in the ASU-FIRST software [10], we calculate these interactions and assign to them an energy using the Lennard-Jones 6-12 potential. Then, as for hydrogen bonds, we use an energy cutoff to determine which interactions to include in the modeling. As a proof-of-concept, we investigated the use of a single, rigid bar to model these interactions. We have implemented these extensions in our KINARI software, and made it available for public use on the KINARI-Web server [21].

To address the need for a fast, effective, evaluation and benchmarking, we propose a method for evaluating the tuning of the hydrogen bond and hydrophobic energy cutoffs. This is an adaptation of the B-cubed score from the information retrieval literature, which is used to compare two clusterings of the same data [2]. We perform an evaluation on a curated data set of proteins whose cluster decompositions were computed by a different method. We have made the benchmarking scripts, written

in python, available at the KINARI web site for public use. This work appeared in ICCABS 2012 and an extended journal version is under submission [24].

1.1.3 Characterizing robustness of rigidity results

We propose a general extension to KINARI that can be used with any modeling scheme. When providing the user with rigidity results on a protein, we can augment this with data on the robustness of the results. What follows is the motivations for this extension and a short summary of the work undertaken to address the problem.

Rigidity analysis is performed on only a single conformation of a protein, typically using coordinates derived from X-ray crystallography experiments. A simplifying assumption in these systems is that the set of interactions identified, particularly the noncovalent interactions, namely hydrogen bonds and hydrophobics, are static constraints. Yet, proteins undergo natural fluctuations around the native state, and molecular dynamics simulations show the noncovalent interactions, *flicker*, breaking and reforming rapidly, typically over nanoseconds [61]. An open question is if the rigidity determined from a single conformation generalizes over the entire ensemble of nearby conformations during fluctuation.

Nearly all PDB files, even those of the highest resolution, contain errors in the coordinates which arise from ambiguities in the experimentally acquired data [9]. The coordinates of the atoms are an estimate of their locations, with uncertainty encoded in the resolution and B-values in the PDB file [74]. Because the occurrence of any particular noncovalent interaction depends on local geometry, the noise in the atom coordinates propagates to the set of interactions identified by software. This was pointed out by Jacobs *et al.* early on, and it was recommended that only the highest quality PDB structures be used for rigidity analysis [48]. Since such high quality data is not always available, it is crucial to understand the effects of noise in the coordinates on the rigidity results.

Understanding the effects of atomic fluctuations and noise in PDB data on rigidity results can be treated as the same problem, posed in the following question.

For the set of conformations near the native state, included in normal protein fluctuations, are the rigid clusters stable?

The problem of understanding the persistence of rigid clusters, over a potentially infinite set of protein conformations, is quite complex. To make progress towards this larger problem, we study a restricted class of ensembles. These ensembles are created not by perturbing atom positions, but by modifying the set of interactions. Observe that although a protein's geometry is needed to calculate important stabilizing interactions, rigidity analysis relies only on connectivity information. Normal protein fluctuations can be simulated by simply modifying the set of noncovalent interactions.

We begin with ensembles created by removing just one interaction at a time. We use these to get information on the original conformation (containing all the interactions). We would like to know the sensitivity of the rigid clusters to small changes in the set of interactions. If any particular interaction within a cluster were to break, would the cluster remain rigid, shatter into many smaller clusters, or would the flexibility increase, but only negligibly?

Contribution. In order to understand how the slight variations in the set of interactions effects rigidity, we propose a method of measuring the redundancy of a cluster. First, we classify each noncovalent interaction as either critical or redundant to the rigidity of its cluster. The counts of critical and redundant interactions can be used to score the redundancy of each rigid cluster. A cluster with a higher redundancy score is less likely to lose rigidity when any interaction breaks. In a 3D visualization of the protein, the clusters are colored by score, so they can be easily compared. In addition, we measure the change in cluster size upon the interaction's removal, which we call the interaction's criticality value. We characterize what is the typical occurrence of redundant and critical interactions with an evaluation on

a benchmark data set of over 120 proteins. We show with case studies that when interactions with higher criticality values (10% or more) are present, they tend to be clustered together around the active site. We make these methods available from the KINARI-Web server (`http://kinari.cs.umass.edu`).

In an earlier form, this work appeared in ICCABS 2011, but has been significantly extended for this thesis [23].

1.2 Thesis outline

This thesis is structured as follows. In Chapter 2, Background and Related Work, we have included relevant background material on protein structure and flexibility, and a short introduction to rigidity theory and the pebble game algorithm. This chapter includes a survey of related work to place the contributions of this thesis in context. Chapter 3 describes the major concepts in KINARI and includes four case studies showing the performance of KINARI v1.0 on 4 proteins which were previously studied with rigidity analysis systems. In Chapter 4, we present our methodology for benchmarking protein rigidity analysis systems. Then, in Chapter 5, we propose new methods for including weaker interactions in the modeling and apply our benchmarking methodology. In Chapter 6, we present our work on evaluating robustness of rigidity analysis results via redundancy analysis.

CHAPTER 2

BACKGROUND AND RELATED WORK

Molecule rigidity analysis is a research area which began as early as the 1980s, combining biochemistry, mathematics, and computation. The history is a very collaborative one, and shows how the interdisciplinary nature of the work has resulted in a method that is elegant, efficient, and applicable to a very important problem: understanding the nature of proteins.

In this chapter, which contains only previous work, we briefly describe the mathematical theory and algorithms underlying rigidity analysis. we also include a short introduction to protein structure. To place the thesis work in context, we survey current laboratory and computational techniques for studying protein motion, including prior studies using rigidity analysis systems. For the interested reader, the last section of this chapter describes the history of the development of protein rigidity analysis, from the 1980s to the present day.

2.1 Primer on rigidity theory and the pebble game algorithm

Flexible and rigid frameworks. A planar bar-and-joint framework is made of fixed-length bars connected by universal joints, with full rotational freedom, Figure 2.1. Only motions which preserve the lengths and connectivity of the bars are permitted. If the framework admits a continuous deformation, permitting the distances between any of the bar endpoints to vary, then it is *flexible*. Otherwise, if no such motions are permitted, the framework is *rigid*.

For example, the triangle framework, depicted in Figure 2.1, is *rigid*. Because each pair of points shares a bar, there is no way to continuously deform the framework in order to change the distance between any pair of points. The four-bar-linkage is flexible because the distance between two points which are on diagonal corners can vary. By placing an additional bar on the diagonal, the framework is no longer flexible, and now forms a single *rigid body*. The framework only permits the rigid body transformations in the plane: translations along the x- and y-axes and rotation. An additional diagonal placed between the remaining two endpoints (not depicted) is an over-constraint; because the framework is already rigid, there is no effect of placing the additional bar. Although the rigidity and flexibility of generic bar-and-joint frameworks can be determined in 2D using simple counting conditions (via Laman's theorem, [62]), the characterization does not extend to 3D.

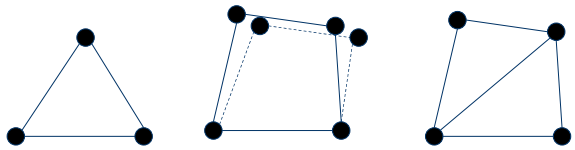


Figure 2.1: Examples of planar bar-and-joint frameworks. The triangle framework is rigid, while a square is flexible. With the addition of a diagonal brace, the square becomes rigid.

Body-bar-hinge frameworks. The focus in this thesis is on a different family of frameworks, for which there are important fundamental theorems supporting their analysis. A *body-bar-hinge framework* consists of rigid bodies connected by fixed length bars and/or hinges [84, 85]. Bars are rigidly affixed to bodies at universal joints. Hinges are a joint between two or more bodies that admit only one motion-rotation about the hinge axis. See Figure 2.2 for examples.

Counting degrees of freedom. In 3D, a rigid body in isolation has 6 degrees of freedom (DOFs): translation along the x, y, and z axes, and rotation around each

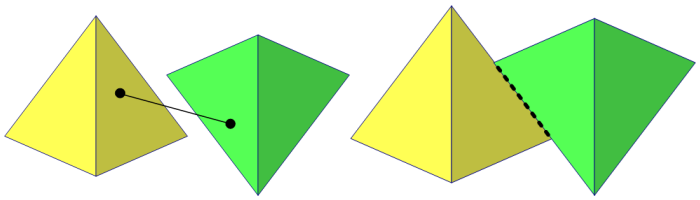
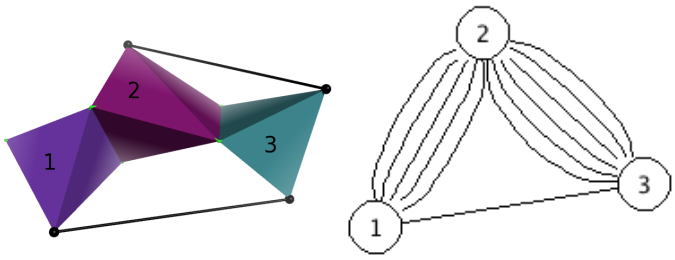


Figure 2.2: Examples of 3D body-bar and body-hinge frameworks. (a) A bar connects the two rigid bodies at universal joints allowing full rotational freedom. This framework has 5 internal DOFs. (b) A hinge joint between two rigid bodies permits only a rotation around the hinge axis. This framework has 1 internal DOF.

of the x, y, and z axes. Two disconnected rigid bodies have a total of 12 DOFs; k disconnected rigid bodies have a total of $6k$ DOFs. When two bodies are connected by a bar, as shown in Figure 2.2 (a), one degree of freedom is removed. Adding additional bars between the two bodies can remove up to 6 DOFs, at which point the two rigid bodies become rigidly attached to one another and form a single rigid body. It is not possible to remove the remaining "trivial" 6 DOFs by placing additional bars or hinge constraints. These 6 DOFs are the rigid transformations of the framework itself. If instead the two bodies are connected at a hinge, Figure 2.2 (b), 5 DOFs are removed. Seven DOFs remain: the 6 trivial DOFs and the one internal DOF, from the rotation permitted along the hinge axis. For any body-bar-hinge framework, each body contributes 6 DOFs. Each bar may remove 1 DOF and each hinge, 5 DOFs.

Tay graph of a body-bar-hinge framework. The associated graph of a body-bar-hinge framework, or *Tay graph* as we refer to it in this thesis, is the multi-graph which contains exactly one vertex for each body, one edge for each bar, and 5 edges for each hinge. Figure 2.3 shows a body-bar-hinge framework and its associated Tay graph.

Tay's theorem. A simple counting rule, due to Tay [84] (see also [86]) and rigorously proven to be valid by *Tay's theorem*, can be used on the Tay graph to determine the rigidity and the DOFs of the framework. A graph is (k, ℓ)-sparse if every subset of vertices in the graph is spanned by no more than $kn' - \ell$ edges. If a graph is (k, ℓ)-



(a) (b)

Figure 2.3: A body-bar-hinge framework and its associated Tay graph.

sparse and has exactly $kn-l$ edges, it is (k,ℓ)-tight. The following theorem describes the relationship between the generic rigidity of the Tay graph and corresponding body-bar-hinge frameworks.

Theorem 1. ***Tay's Theorem****: Theorem for 3D body-bar-hinge (Tay, Whitely): A multi-graph G, with n vertices and m edges, is the graph of a generic minimally rigid body-bar-hinge framework iff any subset of n' vertices in G spans at most $6n'-6$ edges and $m = 6n-6$.*

Pebble game algorithms. A family of pebble game algorithms efficiently analyze graph *sparsity* [64]. For body-bar-hinge frameworks, the $(6,6)$-pebble game played on the Tay graph determines if the framework is generically minimally rigid and if not, what are its rigid components, DOFs, and overconstraints. The pebble game algorithms run in time $O(n^2)$, where n is the number of vertices in the input graph.

We provide only a short description of the pebble game. For further background on rigidity theory and the pebble game algorithm, we refer the reader to the Linkage Lab's educational website (`http://linkage.cs.umass.edu/pg/`) with interactive Java applets and SoCG tutorial video of Lee-St. John, Theran, and Streinu [65].

Figure 2.4 shows an example application of the (2,3)-pebble game algorithm to determine the rigidity of a 2D bar-and-joint framework. The input is an associated graph with one vertex for each joint and one edge for each graph. The (k,ℓ) pa-

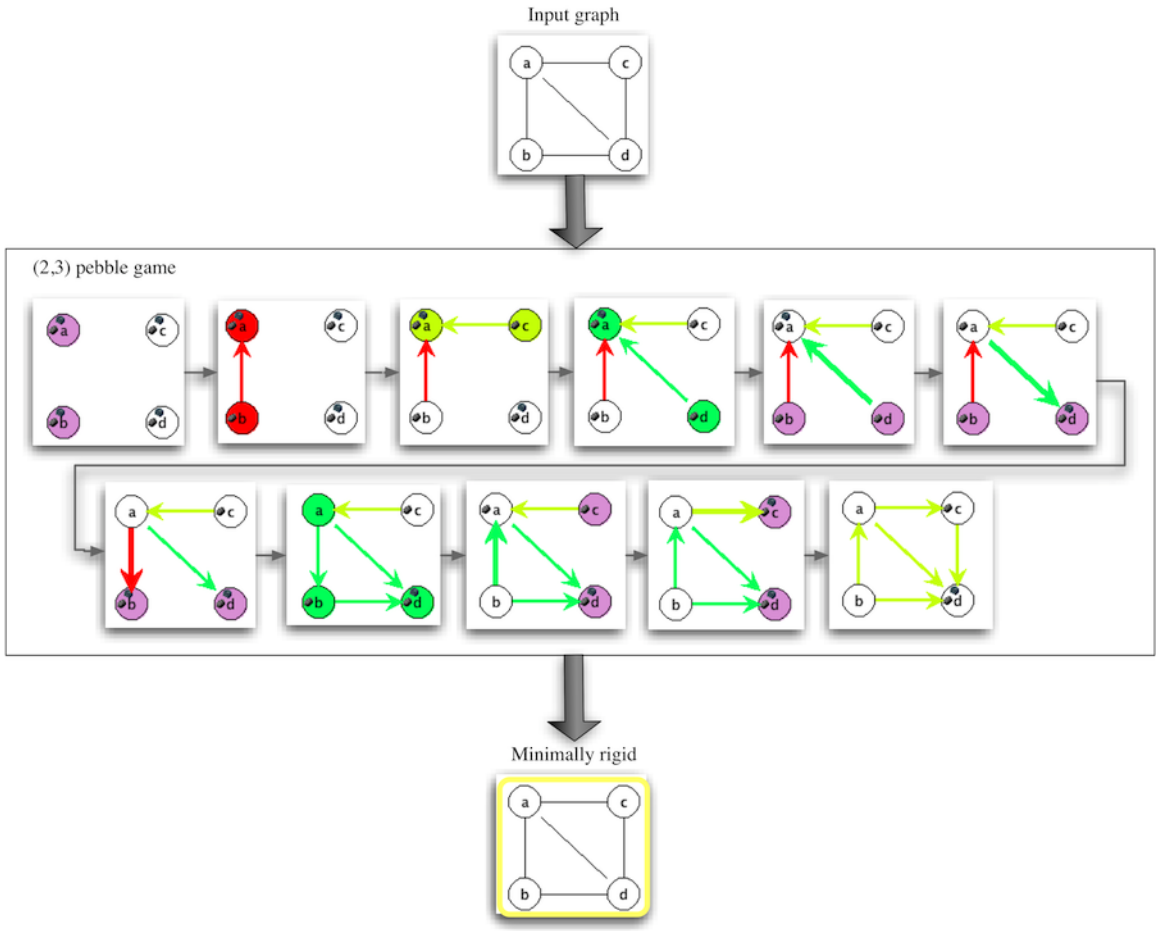


Figure 2.4: Demonstration of the pebble game algorithm for determining rigidity of a 2D bar-and-joint framework. Images generated with Java applet written by Audrey Lee-St. John [65].

rameters are set to $(2, 3)$. A pebble graph is created, with no edges and k (for this example, $k = 2$) pebbles are placed on each edge. Edges are inserted when $\ell + 1$ (for this example, $\ell + 1 = 4$) pebbles can be collected. When an oriented edge is placed into the graph, a pebble is removed from the source vertex. The (k, ℓ)-tight components are detected, and maintained, at each edge insertion. The game is complete when all edges have either been inserted or rejected as overconstraints. The output of the pebble game is the set of components, overconstraints, and the DOFs. Although the worst-case runtime is $O(n^2)$, experiments have shown typical runtimes to be near-linear [46]

2.2 A short introduction to protein structure and flexibility

Proteins consist of one or more polypeptides arranged in a biologically functional way. Each polypeptide is a *chain* of *amino acids* connected together along a *backbone*. There are 20 typically occurring amino acids, each type defined by the *sidechain*. A protein's primary structure is its amino acid *sequence*. Protein backbone forms regular *secondary structure motifs*, namely α-helices and β-sheets, Figure 2.6. Non-secondary structure regions of the backbone are called *loops*. Secondary structures pack together to form the overall *fold* of a chain, also called the *tertiary structure*. A protein may be biologically functional as a single chain, called a *monomer*, or it can be part of a larger complex. The arrangement of multiple chains in a macromolecule is the protein's *quaternary structure*. For example, virus capsids are formed from a large number of proteins assembled symmetrically into a large sphere. Any molecule, protein or not, which binds to another protein is called a *ligand*.

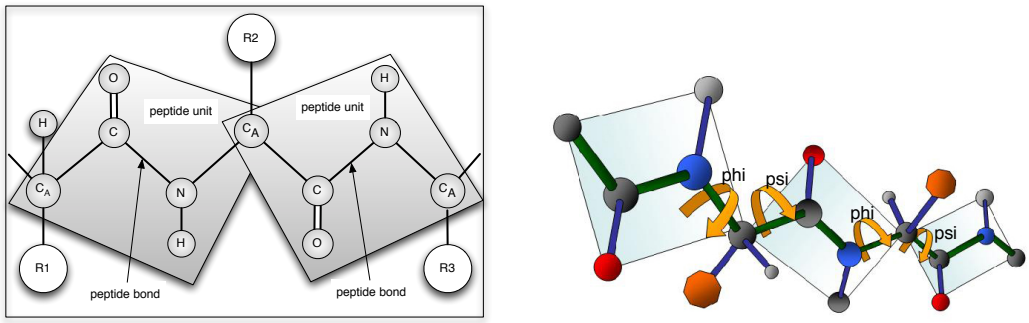


Figure 2.5: The chemical composition of the protein backbone. (a) Each amino acid consists of an NH group, a C_α, a C'=O group, and a sidechain, shown only as R1, R2 and R3. Peptde units are the building blocks of the protein backbone.

Stabilizing bonds and interactions. The protein backbone and sidechains are held together by covalent bonds, formed from the sharing of electrons. Hydrogen bonds are weaker interactions, which essentially form by the sharing of a hydrogen atom. Regular patterns of hydrogen bonding hold together secondary structures, and contribute stability to the fold. The hydrophobic effect, the tendency of certain hydrophobic

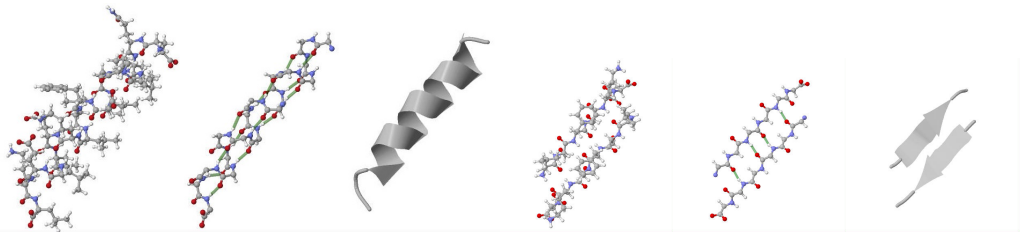
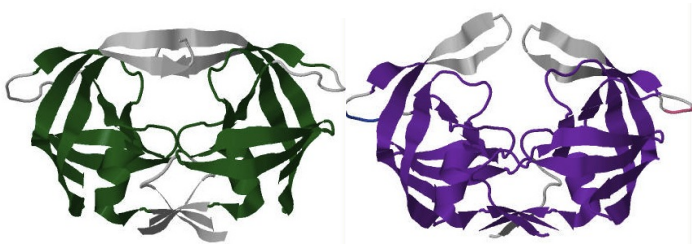


Figure 2.6: α-helices and β-sheets are secondary structure motifs. (a) and (d) show ball-and-stick models of protein fragments which form an α-helix and a β-sheet. The regular hydrogen bond pattern hold the secondary structures together, shown in (b) and (e). Secondary structures are often depicted in cartoon (or ribbon) form (c) and (f). Drawn with Jmol, the helix is 1PEF and the β-sheet is 3LOZ.

residues to avoid water and crowd together to form a hydrophobic core, is the main force behind folding. These hydrophobic interactions also stabilize the protein's fold.

PDB data. Protein 3D structure is determined using X-ray crystallography or NMR spectroscopy. The Protein Data Bank, or PDB, is a free online database containing over 68,000 protein structures. Each structure deposited is assigned a 4-character PDB code, and the data for any of these proteins can be downloaded as a PDB-formatted file [74]. A PDB file contains the type and coordinates for each atom in the macromolecule for which the location was determined.

Protein flexibility. A protein's 3D folded shape determines its function. For example, we examine the protein HIV-1 protease, depicted in the 'closed' and 'open' conformations in Figure 2.7. This protein has been well-studied as a target for drug therapy due to its essential role in the replication of the HIV virus [93]. During the replication process, the protease cleaves a long peptide chain into shorter chains by opening and closing the β-hairpin flaps, like molecular scissors. The two conformations show the two flaps to be quite mobile, while the remainder of the protein remains relatively unchanged. NMR experiments and molecular dynamics simulations have confirmed the flexibility and mobility of the flaps [27, 43].



(a) (b)

Figure 2.7: Two conformations of HIV-1 Protease; the closed conformation (1HVR) and open conformation (1TW7). The upper flaps region open and close in order to perform the cleaving action necessary for replicating the HIV-1 virus.

Graphical rendering of proteins. Many of the figures included in this thesis are renderings of proteins in 3D, mainly generated with Jmol (`http://jmol.org`). Different rendering styles are used depending on the features of interest. The *ball-and-stick* rendering style shows the connectivity of atoms and covalent bonds. For example, Figure 2.6 (a,d) show segments of protein chain which form an α-helix and β-sheet. These motifs involve only backbone atoms, and the sidechains can be stripped off in the visualization to display the atoms and bonds in the backbone only (b,e). The *cartoon* rendering style shows only the shape of the backbone (c,f), permitting easier examination of the secondary structure composition.

2.3 Methods for studying protein structure and motion

Computational methods for studying protein structure and motion rely on high-quality laboratory experimental data. We survey some of the more widely used methods laboratory and computational methods in the next sections.

2.3.1 Laboratory experimental methods

It is not yet possible to directly observe individual atoms moving within the protein. When choosing a technique for studying protein motion, a trade-off must be made between timescale and resolution. X-ray crystallography experiments provide

the highest resolution data, but give a somewhat static picture of the protein at its native, folded state. Alternative laboratory experimental methods, such as fluorescence resonance energy transfer (FRET) or hydrogen-deuterium (H-D) exchange, have been developed for observing motion at lower resolutions.

X-ray crystallography. To resolve the coordinates of a 3D structure of a macromolecule via X-ray crystallography, first the protein must be crystallized. Proteins may still be active in the crystal structure [71]. Each unit cell contains one or more proteins, solvent, and other substrates. X-ray beams are cast through the crystal and a 2D diffraction pattern is collected on a detector. The crystal is rotated in order to collect diffraction patterns at different angles, and then these diffraction patterns are used to determine an electron density map. An iterative process of fitting and refinement is performed in order to resolve the positions of the atoms.

The *resolution* of a PDB file, in units Å, is a measure of precision in the positions of the atoms. In a well-ordered crystal, the atoms will experience less vibration and therefore less noise will be present in the diffraction data. Higher resolution data will have a more detailed electron density map, and there will be less potential for variance in how the molecular model is fit into the density map.

From this density map, the locations of specific atoms are resolved. The structure in the PDB file may contain more than one set of coordinates for some of the atoms, called alternate locations [74]. The resolution, reported in the PDB file, of the determined structure may vary based on experimental conditions. Also, for each atom record in the PDB file, a *B-value* (or *B-factor*) is given, which describes the amplitude of the vibration of the particular atom. This vibration arises from thermal motion or disorder. Figure 2.8 shows the closed conformation of HIV-1 protease (1HVR) colored by B-value. The cooler colors (blue) signify lower B-values and the warmer colors (red) signify higher B-values. The atoms in the hydrophobic core are more packed and tend to have lower B-values, while those side chains on the surfaces

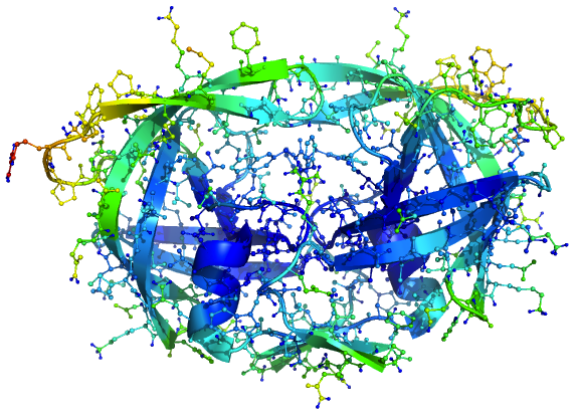


Figure 2.8: Crystal structure of closed conformation of HIV-1 Protease 1HVR) colored by B-value. The atoms are colored from lowest B-value (blue) to highest (red).

have the highest B-values. B-values alone are not sufficient to determine the regions of the protein which will move rigidly and collectively in domain-level motions.

One may assume that it is desirable to use only the highest resolution data available when studying protein flexibility from X-ray crystallography data. However, under such a restriction, only the less dynamic proteins which undergo less conformational motion would be observed. In order to study proteins which undergo larger domain-level motions, one may need to rely on lower resolution data.

NMR spectroscopy. An alternative high-resolution method, NMR spectroscopy examines proteins in solution rather than crystal form. The proteins in solution will undergo some thermal fluctuation, but must be around the global free energy minima, rather than a mix of conformations in different local energy minima. NMR provides the same data on atom coordinates in the PDB file as X-ray crystallography, but in addition, usually includes several models rather than a single structure. Using the technique, a set of distance constraints between geometrically close-by atoms is collected, and then used to determine the positions of the atoms. Because multiple nearby conformations of the molecule agree with the set of constraints, multiple models are included in the PDB file. The different models may describe thermal

motions of the protein in solution. Although these atomic fluctuations are different from conformational motion, the data may provide some insight into the flexibility of the structure, for example in the flaps of HIV-1 protease [27].

Low-resolution methods. Other experimental methods can describe motion under active conditions, as opposed to the steady-state conditions of NMR, but at lower resolution. For example, fluorescence resonance energy transfer (FRET) can be used as a 'spectrosopic ruler' [39]. With this method, two regions of the protein are tagged by attaching fluorescent atoms. The distance between the two fluorescent tags can be measured with time-dependencies. A major limitation of this method is that only a single distance change is measured. Other methods, such as hydrogen-deuterium (H-D) exchange experiments, can be used to measure the rates at which proteins folding, but provide no details of atom positions [39]. For example, H-D exchange experiments have been used to determine the folding order of subregions, known as foldons, of Cytochrome-c [67]. H-D exchange data was used by Rader *et al.* to validate a computational method for predicting folding cores, via a rigidity analysis approach [76, 77].

2.3.2 Computational methods

Computational methods may assist in testing hypotheses in cases where existing laboratory experimental techniques are insufficient.

Molecular dynamics. The most accurate and extensively researched computational method is physics-based simulation, or *molecular dynamics* (MD). Molecular dynamics systems simulate the motions of the atoms as they interact with each other over many small time steps. At each time step, the velocities of the atoms are calculated using an energy equation and the atoms are moved to their new positions [28] (Chapter 5). Some of the major software for performing MD simulations are AMBER [8],

CHARMM [5], and Gromacs [41]. The following excerpt from a 2007 survey paper of Wildman and Kern nicely summarizes the power and limitations of this method:

> Computation has the unbeatable edge in that it can describe protein dynamics completely: the precise position of each atom at any instant in time for a single protein molecule can be followed, along with the corresponding energies, provided that at least one high-resolution structure in known as a starting point... Unfortunately, protein dynamics on the microsecond-to-millisecond timescale is currently out of reach for conventional MD simulations.

Although MD simulations have the potential to simulate the motion of proteins with high accuracy, the scale in simulation time they cover is short, and typically inadequate. Local motions, such as atomic fluctuations, take place on the femtoseconds (10^{-15}) to milliseconds (10^{-6}) timescale. Rigid body motions, such as protein domain motions, take place on the nanoseconds (10^{-9}) to seconds timescale. Large scale motions, such as folding and unfolding, may take place on the milliseconds to seconds timescale. To achieve a reasonable amount of accuracy, MD simulations use a time step somewhere between a femtosecond (10^{-15}) and a picosecond (10^{-12}). Using MD, simulating the folding of a 100 residue protein is not yet possible. The holy grail of these projects is to simulate the complete and correct folding of a protein, with no human intervention or information on the target conformation. The Folding@Home project has successfully simulated the millisecond-timescale folding of a 32 residue protein by using the same approach as SETI@HOME, using free clock cycles of volunteers' personal computers during unused periods [92]. The first millisecond timescale MD simulation on a 58 residue protein required over two months to compute on the Anton supercomputer in 2009 [57].

Monte Carlo. Unlike MD, Monte Carlo simulations do not produce time-scale accurate simulations, but instead aim to sample a much larger area of conformation

space [28]. Monte Carlo is a method for collecting an entire ensemble of possible protein conformations. The method takes as input one initial conformation of a protein, then perturbs the positions of the atoms at random. The energy is computed using the same molecular mechanics forcefields used by MD, and the conformation is either accepted or rejected, with some probability, based on the energy. This sampling process is repeated until the space of conformations has been sufficiently sampled. Monte Carlo experiments explore the energy landscape more thoroughly and efficiently than MD, but do not produce accurate trajectories of protein motion.

Normal Mode Analysis. Normal Model Analysis (NMA) is another standard modeling technique for molecular mechanics. In NMA, the harmonics of the molecular system are studied, with the focus on characterizing the lowest frequency harmonics which may be associated with the larger scale motions. The method detects subsets of atoms which move "collectively" in the model [13]. Rather than generating different sets of coordinates for the atoms in a protein, NMA attempts to identify domains in a protein which vibrate at the same frequencies. NMA detects the *mobility* of atoms in the molecule by analyzing correlated motions, but does not determine information on the rigidity and flexibility of structural regions. NMA may show that atoms in a certain domain, like a mobile α-helix, move collectively, but it won't show that this region is rigid and unlikely to deform, while other regions, which are flexible, will permit deformation.

Rigid cluster decompositions using multiple conformations. A number of methods have been proposed for determining a rigid cluster decomposition from two conformations of a protein, such as HingeFind [97], DynDom [66], and RigidFinder [1]. All of these methods aim to find clusters of residues where the differences between pairwise distances in the two conformations falls within some error tolerance. RigidFinder uses a dynamic programming approach in order to do this. The RigidFinder method was validated on a number of case studies, comparing the method with experimen-

tal data and with prior methods. In the next three chapters of this thesis, we refer to this RigidFinder data set and associated decompositions in order to measure the performance of KINARI at determining rigid cluster decompositions.

2.4 Prior work using protein rigidity analysis systems

Prior software systems. The first application of pebble game rigidity analysis to molecular system was a study of glass networks in 2D by Jacobs and Hendricksen [46] and Jacobs and Thorpe [49]. In these studies, rigidity percolation of glass networks was studied by modeling the molecular system as a bar-and-joint framework and applying the 2D pebble game. By randomly "diluting" the system of bonds, they effectively modeled a phase transition of glass between a solid state and a liquid state.

Later, Jacobs extended the pebble game algorithm to a 3D bar-and-joint version [45]. The explanation of the algorithm did not have a proof of mathematical correctness, but experimental validation of the algorithm was performed by comparing the rigid clusters identified by the algorithm with those identified by numerical analysis. This was then applied to study proteins [48] using the software FIRST. This version of FIRST (recently renamed Proflex), which we refer from this point on as MSU-FIRST (as it was developed at Michigan State University), uses Jacobs' 3D bar-and-joint pebble game. A rigorous study of Jacobs' 3D pebble game by Chubyinksi and Thorpe later demonstrated that although the heuristic method frequently produced correct results, the errors were unpredictable [11].

MSU-FIRST was later upgraded to a version which used body-bar modeling of the proteins and ran the 3D body-bar pebble game [10]. We refer to this version, developed in Thorpe's laboratory at Arizona State University, as ASU-FIRST. The rationale presented for the move from the bar-and-joint modeling to the body-bar modeling is that there were complications in modeling non-covalent interactions, specifically

the hydrophobic effect, in the bar-and-joint model that could be "overcome" in the body-bar model.

Validation of protein rigidity analysis software. A number of techniques were used to validate that the MSU-FIRST software produced correct results. We survey here the techniques used to validate.

Comparing rigid cluster decompositions of unique conformations. In the 1999 book chapter which introduced the MSU-FIRST software, a few types of validation were performed [47]. First, the DOFs counts for a few very small examples computed by FIRST were shown to be correct; the small examples used were single α-helices of different lengths and a non-biological molecule called cubane. Then a validation on a real protein was performed, by investigating the results produced by the software for two conformations of the same protein, a lysine-orthinine-arginine-binding (LAO-binding) protein, in the open (2LAO) and closed (1LST) states. Running the software MSU-FIRST on both conformations, they found that "Most of the rigid substructures and underconstrained regions identified in the two conformations correspond to one another". For the two conformations, a *flexibility index* for each bond is assigned. If a bond is in an overconstrained region, meaning it lies in a rigid cluster with redundant bonds, as determined by the pebble game, the flexibility index is the fraction of redundant bonds to all bonds in a rigid cluster. If a bond is not in a rigid cluster, the flexible region it is in is computed, and the flexibility index it is assigned is the fraction of rotatable bonds to the number of DOFs in the region. Using this flexibility index, they were able to reason about some of the expected differences in flexibility between the two conformations which "correlated with known motions". A 2001 journal paper extended the validation [48], including 3 new case studies in the flavor of the LAO-binding protein study. Each protein examined had multiple structures from unique conformations deposited in the PDB; these were HIV1-protease (1HHP, 1HTG); adenylate kinase (1AKY; 1DVR), and Dihydrofolate

Reductase (1RA1, 1RX1, 1RX6). A thorough analysis was provided, comparing their own proposed residue-based flexibility index with PDB B-values, RMSD values, and changes in Φ and Ψ angles. The cluster decomposition itself was analyzed qualitatively, by comparing known flexible domain-level hinge regions with those identified by the software. In order to validate that the results of KINARI match with those previously published, we will revisit these case studies in Chapter 3, Section 3.4.

Comparing rigid cluster decompositions of protein homologues. A later study did not attempt to compare the rigidity results between two conformations, but instead verified the similarity of the rigid cluster decompositions of protein homologues. MSU-FIRST analysis was performed on three different Cytochrome-c proteins from three different species: horse (1HRC), yeast (1YCC), and a bacteria (1CO6_A) [90]. In the study, "strong similarity in their flexible regions" was found in the results between the sequence-aligned proteins. In addition, in a dilution analysis on the three proteins (described next), the changes in the backbone rigidity in the three dilutions were found to agree.

Validating dilution (simulated unfolding). The bond dilution technique has been applied to protein folding; when the hydrogen bonds are removed (or "broken") by order of energy, it is called simulated unfolding [78]. They were able to find a correlation between the "mean coordination" of the protein, $< r >$, with the "fraction of floppy modes", $f = F/3N$, where F is the number of DOFs and N is the total number of atoms. That is, before bonds are placed, a protein has 3 DOFs for each atom, so the "fraction of floppy modes' is the fraction of DOFs over the total number of DOFs. They correlate $< r >$ with the first derivative of f to find the critical value where a phase transition occurs from rigid to flexible. This is hypothesized to be the same transition from the folded to unfolded states of the protein.

The simulated unfolding technique has been applied in the identification of protein folding cores [76, 77]. A protein folding core is defined to be the region of the

protein that initializes folding, and it is hypothesized that the folding core is the region of the protein that rigidifies first during folding and loses rigidity last during unfolding. The study tests whether dilution experiments can find the folding cores. The rigid groups of atoms in the protein's backbone are monitored during the simulation. Experimentally detected folding cores are compared with the largest rigid groups that remain. Folding cores identified by dilution were found to agree with the experimentally determined folding cores of 10 proteins: BPTI, 1BPI; Ubiquitin, 1UBI; CI2, 2CI2; Ribonuclease T1, 1BU4; Cytochrome c, 1HRC; Barnase, 1A2P; α-Lactalbumin, 1HML; Apo-Myoglogin, 1A6M; Interleukin-β, 1ILB; T4 Lysozyme, 3LZM [40].

Limited experiments were performed with orderings of hydrogen bonds other than by energy. The two evaluated were (1) choosing one of the 10 lowest energy bonds at random and (2) choosing a bond completely at random. They found the first method was able to identify the folding core just as well as the non-randomized ordering. The second method, completely randomized, could sometimes identify the correct folding core, but sometimes could lead to a completely incorrectly identified folding core. Thus they concluded that the energy of the hydrogen bonds is a significant factor in simulating unfolding. This study was performed on 10 proteins.

Hinge prediction. The StoneHinge method and server for domain-level hinge prediction, developed in the Gerstein Lab, used the MSU-FIRST software as a module [56]. In their study they concluded that the MSU-FIRST-based method overpredicted the occurrence of hinges when compared with a set of literature-annotated hinges. The StoneHinge developers resorted to a consensus-based approach, combining the rigidity analysis results with another independent method, in order to achieve better precision in their predictions. Their data set was composed of: CAPK, 1CTP, 1ATP; Bence-Jones protein, 4BJL; LAO-binding protein, 2lao, 1lst; adenylate kinase, 2ak3, 1ake; glutamine binding protein, 1GGG, 1WDN; DNA Polymerase β,

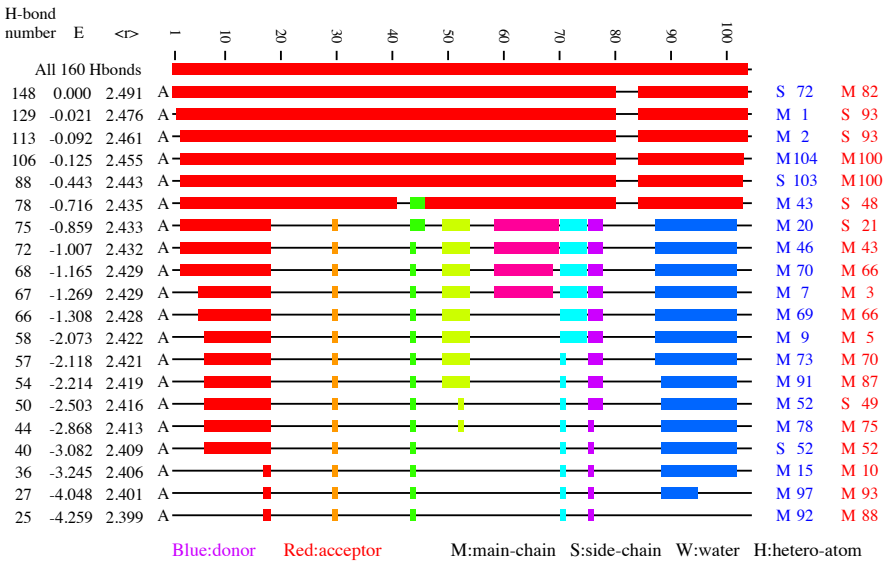


Figure 2.9: A 2D dilution plot produced by ASU-FIRST on Cytochrome-c (1HRC). Each line shows the backbone rigidity, with each cluster assigned a different color.

2BPG, 1BPD; Calmodulin, 1CFD, 1CLL; Inorganic Pyrophosphatase, 1K23, 1K20; ribose binding protein, 1URP, 2DRI; Ig domain of protein G, 1PDB; hydropterin pyrophosphokinase, 1HKA; Cyclophilin A, 1BCK; Rhizopuspepsin, 2APR, 3APR; Chloramphenicol Acetyltransferase, 2CLA, 3CLA; and Proteinase A, 2SGA, 5SGA.

Corresponding states of flexibility in protein homologues. Another study that sought to validate dilution as a simulated unfolding built evidence toward the "corresponding states of flexibility" hypothesis [33]. The hypothesis is that mesophilic and thermophilic enzymes are in corresponding states, with similar flexibility characteristics, at their respective optimal temperature. In this study, they collected a dataset of 20 pairs of homologs and plotted the cluster configuration entropy and the rigidity order parameter during the simulated unfolding and used different criteria to identify the phase transition temperature. They found that for two-thirds of the proteins, the phase transition temperature was higher in the thermophile than in its mesophilic counterpart. A more recent study by the same authors strengthened the validation, relating the corresponding states detected to the protein's activity [79].

Variations in hydrogen bond and hydrophobic interaction network. MD simulations show that noncovalent interactions, the hydrogen bonds and hydrophobic interactions, *flicker* on and off, breaking and forming at varying rates, but typically over nanoseconds [61]. The *duty cycle* is the percentage of time a certain interaction exists during the simulaton. Duty cycles for different hydrogen bonds and hydrophobic interactions were computed from MD simulations, and their duty cycles used as a criterion for inclusion in a rigidity analysis experiment. They performed experiments on two different proteins (barnase and the glutamate receptor ligand). This flickering phenomenon contributes evidence to the need for studying the tolerance or rigidity analysis in a protein to changes in the hydrogen bond set.

Heuristic-version of pebble game to predict ensemble rigidity. More recently, the Jacobs' lab proposed a heuristic method, called the *virtual pebble game*, for predicting ensemble-averaged rigidity for a protein with fluctuating noncovalent interactions. To validate the method, the Rand Measure, which is a count of the number of items that match between two decompositions, was used to analyze a data set of 272 proteins with 3 domains or fewer from the SCOP database [35].

Course grained models for motion generation. Another application of rigidity results has been in motion generation. The FRODA method was included in ASU-FIRST to generate motions by moving rigid clusters and maintaining chemical constraints [95]. Thomas *et al.* have used rigidity analysis as a module in the probabilistic roadmap method, in order to provide a course-grained representation of the protein for more efficient conformation sampling [87]. The NMSim server, recently released by the Gohlke lab, combines a rigid cluster decomposition and low-frequency normal modes from normal mode analysis to generate motions [60].

2.5 Historical overview of development of rigidity analysis theory and applications to molecules

We give some historical perspective on the development of graph-based characterizations and algorithms for studying the rigidity and mobility of molecules.

The 1982 meeting, and subsequent volume, *Symmetries and properties of non-rigid molecules*, was the first international symposium on important developments concerning the symmetries, topologies and properties of flexible molecules. The volume contains the first conjecture (the *Dress conjecture*) on the rigidity of 3D bar-and-joint framework models of molecules, posed by Dress, Dreiding and Haegi [16] and discussed in [86]. Laman's theorem [62], for 2D bar-and-joint frameworks, gives a combinatorial characterization for determining rigidity. Although the equivalent characterization for 3D bar-and-joint frameworks is necessary, it is insufficient for determining whether or not a framework is rigid. The Dress conjecture was an effort to extend Laman's characterization to 3D bar-and-joint frameworks.

Although a combinatorial characterization for generic 3D bar-and-joint frameworks is still an open problem, a different rigidity model, the body-bar-hinge framework, was found to have a Laman-type combinatorial characterization. The theorems for body-bar (Tay, 1981) [83] and body-hinge frameworks (Tay, Whitely, 1984) [86] are the foundation for mathematically-guaranteed graph-based rigidity analysis of molecules. In this thesis, we refer to the combination of these theorems, for body-bar-hinge frameworks, as *Tay's theorem*. With the theorem for body-hinge frameworks, Tay and Whitely posed the *molecular conjecture* for body-hinge models of molecules, which is that Tay's theorem for generic body-hinge structures could also be applied to molecules, which have non-generic configurations. 25 years later in 2009, Katoh and Tanigawa produced a proof of the molecular conjecture [54, 55].

Laman's and Tay's theorems give combinatorial characterizations, but either of these theorems, implemented directly, result in exponential-time algorithms. In 1980,

Sugihara proposed a polynomial-time graph algorithm to test the independence of set of edges for the graph of a 2D bar-and-joint framework [81]. Improving on this in 1985, Imai proposed an $O(n^2)$ network flow algorithm for 2D bar-and-joint graph rigidity testing and extracting the maximal rigid component [44]. Gabow and Westermann proposed an $O(n^2)$ algorithm using matroid sums [30].

In his 1991 Ph.D. thesis, Hendrickson proposed an $O(n^2)$ algorithm using bipartite matching that could not only test if a graph was rigid or not, but also detect all the rigid and redundantly rigid components. Hendrickson's thesis addressed the *molecule problem*, the problem of determining the coordinates of a set of points in space from a set of pairwise distance measurements. One motivation behind the problem is determining molecular structure from NMR spectroscopy data. Rigidity analysis, and in particular, redundant rigidity where the removal of any constraint in a rigid component will have no effect on the rigidity, are applied toward this problem. The thesis contains combinatorial and numerical approaches toward solving the molecule problem, and a description of the ABBIE software package for identifying regions of a molecule for which the coordinates can be determined [38].

In 1995, Franzblau proposed an $O(n^2)$ algorithm, based on a technique called chain- or ear- decomposition, to compute a non-trivial lower bound on the of a 3D bar-and-joint framework with fixed angle joints [25]. She followed this with a variant of the algorithm to compute a non-trivial upper bound in 2000 [26]. Although Jacobs and Thorpe were using an implementation for experimental studies of glass networks as early as 1995 [49], the first paper on Jacobs' and Hendrickson's pebble game algorithm for 2D bar-and-joint frameworks was published in a physics journal in 1997 [46]. The pebble game was a more practically implementable version of the bipartite matching component finding algorithm of Hendrickson's thesis. In 1996, Moukarzel proposed a bipartite matching method for determining rigidity in 2D body-bar structures [70]. In 1997, Jacobs' proposed a heuristic variant of the

pebble game algorithm, for determining rigid components in 3D bar-and-joint graphs of molecules [45]. In 2005 (with a journal publication in 2008), Streinu and Lee proposed a generalized version of the pebble game algorithm to determine properties of (k, l)-sparse graphs [64].

In order to make rigidity analysis techniques available for research on proteins and other molecules, a few different software tools have been published. The bar-and-joint modeling and heuristic pebble game algorithm of Jacobs were implemented in the patented MSU-FIRST software [48, 50]. Later, the MSU-FIRST software was upgraded to the ASU-FIRST software and Flexweb server (`http://flexweb.asu.edu`); the bar-and-joint modeling was switched out for a body-bar-hinge type model which could rely on Tay's theorem and the (6,6)-pebble game [10, 89]. In 2005, the RIGIX software was introduced which uses a variant of the bar-and-joint modeling and Jacobs' heuristic pebble game [15]. The PRM rigidity module, using yet another body-bar-hinge-type modeling scheme, was published in 2007 [87]. Our system, KINARI and KINARI-Web server (`http://kinari.cs.umass.edu`) relies on body-bar-hinge modeling [21].

CHAPTER 3

KINARI: SOFTWARE FOR RIGIDITY ANALYSIS, WITH APPLICATIONS IN MOLECULAR MODELING

We have developed the KINARI software suite and libraries as a research tool for studying protein rigidity and flexibility. The software is the first to support accurate body-bar-hinge mechanical modeling of molecules. In this chapter, we describe the concepts and design of KINARI, and provide case studies of its application on real proteins.

The chapter begins with a review the justifications for building the KINARI library, in Section 3.1. Details on the main concepts in KINARI are included in Section 3.2. This section includes the parameter settings used in KINARI v1.0, to which we will refer in the later chapters of this thesis. These settings are required for full reproducibility. In Section 3.3, we describe the software design and features. In order to demonstrate the use of protein rigidity analysis software to describe protein behavior, Section 3.4 contains four case studies comparing the results on KINARI with those published for MSU-FIRST [48].

3.1 Motivation: mechanically accurate modeling for protein rigidity analysis

Prior software systems, namely MSU-FIRST [48] and ASU-FIRST [10], were developed as a proof-of-concept, in order to demonstrate the usefulness of rigidity analysis in real protein studies. But the approaches taken in these systems had some limitations.

For 3D, the only class of mechanical model for which a combinatorial characterization exists is the body-bar-hinge framework, and therefore, in order to provide correct and precise rigidity results, a molecule must be properly modeled as such. MSU-FIRST used a different type of underlying model, the bar-and-joint model, and because of this, needed to resort to a heuristic version of the pebble game. An extensive study of Jacobs' 3D bar-and-joint pebble game was performed by Chubynsky and Thorpe, with the conclusion that, although the algorithm often produces correct results, it is approximate and there are frequently errors in the rigid cluster decomposition and degrees of freedom determined [11]. ASU-FIRST was developed to overcome this major deficiency, relying on a body-bar rigidity model. Unfortunately, the modeling implementation of ASU-FIRST was imprecise. It mixed the mechanical modeling and graph representation, and did not support the notion of a hinge.

Figure 3.1 shows how KINARI differs from ASU-FIRST. KINARI builds a mechanical model where rigid bodies of atoms overlap on rotatable bonds that behave as hinges, as shown in Figure 3.2. In contrast, ASU-FIRST models the protein directly as a multi-graph where vertices represent groups of atoms and each edge represents the removal of a single degree of freedom between the atoms. ASU-FIRST's rigid clusters are disjoint; they do not overlap at hinge joints. The rigid clusters identified by KINARI and ASU-FIRST are not identical, but when the same input PDB files, bonds and interactions, and modeling options are used, they will be in one-to-one correspondence. On-going work on KINARI investigates extensions that will further increase the modeling accuracy.

3.2 Key concepts in KINARI

Rigidity analysis decomposes a protein into rigid clusters. A *rigid cluster* is a maximal set of atoms held together by interactions determining all the inter-atomic

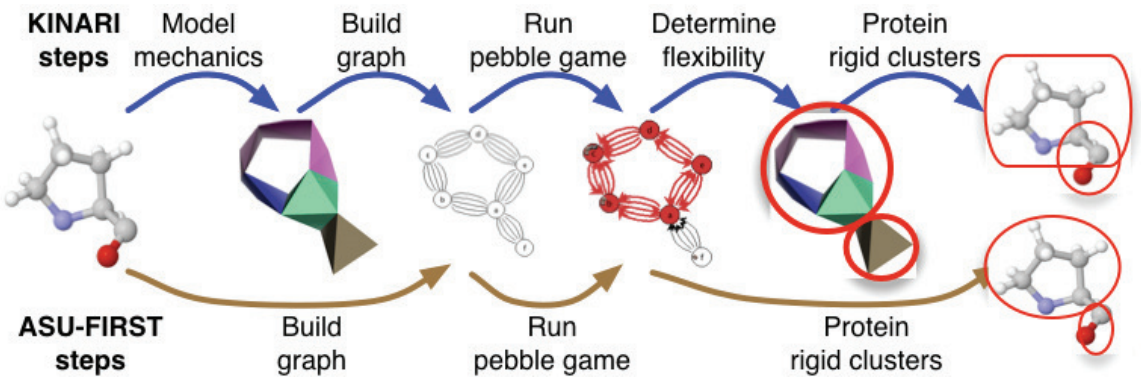


Figure 3.1: Comparison of steps performed by KINARI and ASU-FIRST. A molecule is modeled as a mechanical structure called a body-bar-hinge framework, from which an internal multi-graph is built. The pebble game algorithm calculates components in the multi-graph, from which the flexibility of the mechanical structure and protein rigid clusters are inferred. A prior system, ASU-FIRST, calculates the graph directly from the molecular data, and returns clusters that are disjoint sets of atoms.

distances within the cluster. For small molecules, we can determine the rigid clusters by hand. For example, we can identify two rigid clusters in the proline molecule, Figure 3.2. One cluster is formed by the five atoms in the ring, known to be rigid, and all of the atoms bonded to these 5 atoms. The other cluster is formed by the two atoms bonded by the red-colored bond, and the last tail atom.

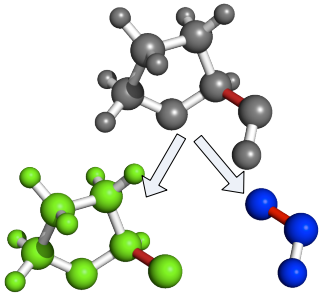


Figure 3.2: Rigid cluster decomposition of proline. A rigid cluster is a maximal set of atoms and all bonds and interactions that hold them rigidly together. The proline molecule (grey) is composed of two rigid clusters. The first rigid cluster (green) is composed of the 5 atoms forming a ring, and all of their covalent neighbors. The one atom not contained within this first cluster, plus the two atoms that share the red bond, form the second cluster (blue).

KINARI uses a mechanical modeling approach to determine the rigid clusters. Figure 3.1 shows the steps performed by KINARI for running pebble game rigidity analysis on a molecule. The input is a PDB file containing the set of atoms in a protein with coordinates. The output is the maximal rigid clusters of atoms, as well as the hinges and bars between the clusters. The first step is *curation* – the PDB file is parsed and the relevant atoms, bonds, and interactions are calculated. After curation, the protein is modeled as a body-bar-hinge framework. From the body-bar-hinge framework, a Tay graph is built and then the $(6, 6)$-pebble game is run to determine the rigid components of the graph. These components are used to build a minimized body-bar-hinge framework model of the same protein, where each body is now maximal. Information from the minimized body-bar-hinge framework is then mapped back to the protein data, in order to determine the rigid clusters of atoms in the protein.

In the next section we describe the curation process (Section 3.2.1) and in Section 3.2.2, we give important details of how modeling is performed. In Section 3.2.2.2 we've included the procedure to convert atom-level clusters produced by KINARI to a residue-level representation. We will make use of the residue-level representations in order to compare results from our system with other available protein rigidity data in Chapters 4 and 5.

3.2.1 Curation

Before modeling is performed, the PDB data must be processed to include only macromolecular data that is relevant for the rigidity analysis experiment. We refer to the processing of PDB data, including calculation of relevant bonds and interactions, as *curation*. The four steps of curation, shown in Table 3.2, were designed collaboratively under the lead of Ileana Streinu. The code for the curation executables in KINARI was written by Filip Jagodzinski, who extended the steps in his thesis

studies of rigidity of the biological unit and crystal forms. In order to allow others to reproduce our results, we describe the details for each step. The default curation settings of KINARI v1.0 are listed in Table 3.2.

Step	Description
1	Specify models, chains, ligands and water molecules to retain.
2	Remove alternate atoms and add hydrogen atoms.
3	Calculate chemical bonds and interactions.
4	Prune undesired interactions or add custom chemical constraints.

Table 3.1: Curation steps to prepare protein data for rigidity analysis. Chains, ligands, or chemical interactions can be removed or added to determine their effect on protein rigidity.

3.2.1.1 Curation step 1: Specify models, chains, ligands and water molecules to retain

In the first step, the user selects the parts of the macromolecule contained in the data file to be retained for rigidity analysis. For example, PDB file 1HVR, one of the available X-ray crystallography data files of HIV-1 Protease, contains two chains, A and B, as well as a ligand, XK2. A user may choose to analyze the flexibility of 1HVR with and without the ligand in order to observe the effects on rigidity with the ligand's inclusion. Currently KINARI only supports PDB files with 100,000 atoms or fewer. This is due to limitations in the column widths in the PDB format [74].

Step	Description
1	Retain all chains, remove all ligands and waters.
2	Retain only first occurring alternate atom. Add hydrogens with Reduce.
3	Calculate hydrogen bonds with HBPLUS and hydrophobic interactions with heuristic.
4	Prune nothing.

Table 3.2: Default curation parameter settings in KINARI v1.0

3.2.1.2 Curation step 2: Remove alternate atoms and add hydrogen atoms.

X-ray crystallography PDB files do not contain hydrogen atoms, which participate in forming hydrogen bonds, the critical elements in stabilizing the biomolecule. Step two of the curation process uses the *Reduce* software to insert hydrogen atoms into the PDB file [96]. PDB files frequently contain multiple alternate locations for the same atom. In order to resolve these ambiguities, by default only the first alternate atom position is retained. A user wishing to retain different positions would edit their PDB file directly.

3.2.1.3 Curation step 3: Calculate chemical bonds and interactions, and assign energies.

In the third step, important stabilizing interactions are calculated, including covalent bonds, resonance bonds, disulfide bonds, hydrogen bonds, and hydrophobic interactions.

Single and double covalent bonds, resonance bonds, and disulfide bonds. Covalent bonds are formed by the sharing of electron atoms between atoms. Covalent bonds are the strongest bonds that exist in nature. Once a protein is formed, we assume that none of the covalent bonds will break. Once two atoms are covalently bonded, the bond length is fixed. In addition, the bond-bending angle between each pair of atoms covalently bonded to the same atom is fixed. In general, covalent bonds permit rotation around the bond, so that the dihedral angle is varied.

There are some special cases of covalent bonds where this rotation does not occur. Double covalent bonds, where multiple electrons are shared between two atoms. These bonds restrict rotation around the bond axis, fixing the dihedral angle. Resonance bonds, for instance peptide bonds, have a partial double covalent bond character. The C-O bond in the peptide group, from which an electron is delocalized and contributed

to the C-N peptide bond, is classified as a single covalent bond. This is to allow rotation around the bond specifically when the O serves as a hydrogen bond acceptor. This follows the convention originally set in MSU-FIRST [47].

A disulfide bond (or bridge) is a strong bond that forms between the sulfurs in two cysteine side chains. These bonds are known add stability to the protein's folded structure. For example the protein 58 residue bovine pancreatic trypsin inhibitor (BPTI, PDB 1bpi) contains 3 disulfide bonds. When these 3 disulfide bonds are experimentally eliminated, the protein is known to unfold [75] (page 28). The disulfide bonds are listed in the PDB file as an SSBOND record [74].

KINARI determines single covalent bonds, double covalent bonds, and resonance bonds in the backbone and sidechains of the 20 typically occurring amino acid types. For modified residues or ligands, a CIF-format file, containing connectivity information, must be supplied [6]. The CIF files are easily retrieved from the PDB website via the code listed in the PDB file under the residue name column.

Hydrogen bonds. A hydrogen bond is the attractive force between one electronegative atom and a hydrogen covalently bonded to another electronegative atom. The electronegative atom is generally nitrogen or oxygen. A hydrogen bond is much weaker (-15kcal/mol or weaker) than a covalent bond (around -85 kcal/mol). The bond strength depends on temperature, pressure, bond angle, and environment (usually characterized by local dielectric constant).

KINARI uses the HBPLUS software to identify hydrogen bonds [69], and assigns energies to them using the Mayo energy function [68]. An energy cutoff can be used to reject the weaker hydrogen bonds and only include the stronger bonds. Generally, hydrogen bonds range in strength between 0 and -7 kcal/mol.

Hydrophobic interactions. As a protein folds, its hydrophobic side chains tend to pack into the interior of the protein, creating a hydrophobic core and a hydrophilic surface. This behavior is thought to be the main driving force for folding. A hy-

drophobic interaction between two atoms is the tendency for these atoms to remain near each other. KINARI provides two different methods for identifying hydrophobic interactions.

The first is a heuristic method, reproducing the H3 function described in the ASU-FIRST manual [88]. The heuristic finds all S-S, S-C, and C-C atom pairs such that (1) the surfaces of the two atoms are within a cutoff distance of each other AND (2) the number of covalent bonds between the two atoms is more than 3. The default surface cutoff distance is 0.25 Å. The default radii for C and S are 1.7 and 1.8. This heuristic aims to distribute the hydrophobic interactions evenly without placing too many constraints too close together, which may completely rigidify a region.

The second method, implemented first in the KINARI software, uses interactions as calculated by a molecular mechanics forcefield, namely, the potentials from the Amber99 forcefield [8]. Unlike the heuristic method introduced in ASU-FIRST, the atom types involved in the hydrophobic interaction are not restricted, so N and O atoms are included. H atoms are not included because they already contribute to structural stability through hydrogen bonding.

To determine hydrophobic interactions in a macromolecule with either method (heuristic or van der Waals potential approach), all pairs of atoms that fall within some cutoff distance must be determined. In a naive implementation, the distance between every pair of points is calculated, requiring $O(n^2)$ time.

In order to speed up to this process to linear time, we preprocess the atoms into a grid data structure. First, the bounding box of the set of atoms is calculated, by determining the maximum and minimum for all x, y, and z coordinates. Then the cell width is set to the cutoff distance and a grid data structure is initialized with a 3D array. The grid is populated with atoms, and the grid cell index for each atom stored in a table for quick lookups. This preprocessing step is performed in linear time.

The number of distances that need to be computed for each atom is now decreased from $O(n)$ to a constant. For each atom, only the distances between that atom and each of the atoms within its grid cell and all adjacent cells need by computed. Because the van der Waals radii bounds the number of atoms that can fit in any grid cell, there is a bound on the total number of atoms for which distances need be computed.

We assign an energy based on the van der Waals energy, using the Lennard Jones 6-12 potential, parameterized with values from the Amber 99 forcefield. This is a new feature that implemented as part of the work in thesis on energy-refined modeling. We will describe this further in Chapter 5, Section 5.3.3 of this thesis.

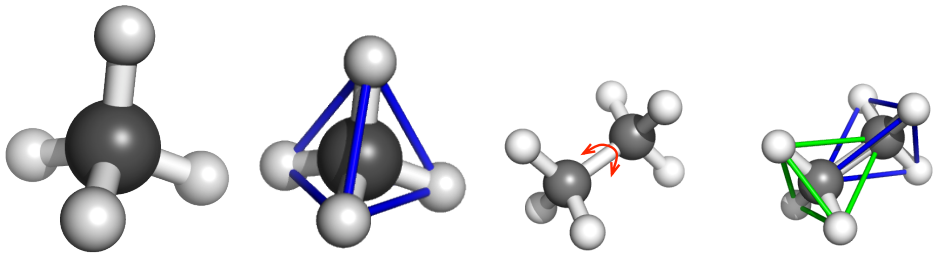
3.2.1.4 Curation step 4: Prune undesired interactions or add custom chemical constraints.

In the final curation step, the computed chemical interactions that exist between atoms in the PDB-formatted input file are presented to the user, who can designate which of them should be retained, and which should be removed. In the case of covalent, resonance, disulfide, or hydrogen bonds, a user can remove constraints within a certain energy range or below or above a certain cutoff value. Chemical interactions that KINARI did not detect but which should be included in the molecular model of the protein, can be easily supplied as user-defined constraints. The ability to manually add chemical constraints is a novel feature; it can be used for formulating rigidity-based hypotheses regarding their effect on the stability of the protein.

3.2.2 Modeling molecules as body-bar-hinge frameworks

Once the relevant atoms, bonds, and interactions have been collected, the protein is modeled in preparation for the pebble game algorithm, as illustrated in Figure 3.1. In this section, we describe how the mechanical model is built.

The default style models the mechanics of the protein as a body-bar-hinge framework. A *body* is a set of atoms rigidly attached to each other, as determined by



(a) Methane (b) Tetrahedron (c) Ethane (d) Two tetrahedra

Figure 3.3: Converting a molecule to a body-bar-hinge framework. Methane (a) is rigid because all pair-wise distances between atoms are fixed (b). In ethane (c), each carbon atom (gray) and its bonded neighbor atoms form a rigid cluster. The two clusters share a hinge along the center C-C bond. The abstract body-bar-hinge framework for ethane is shown in (d); two rigid bodies (represented as tetrahedra) share a hinge along the rotatable bond.

constraints imposed through chemical bonds and stabilizing interactions. For example, methane CH_3, the molecule in Figure 3.3a, is rigid, because all the pair-wise distances between atoms are determined by the covalent bond length and angle constraints. Abstractly, this can be visualized as the rigid tetrahedron from Figure 3.3b.

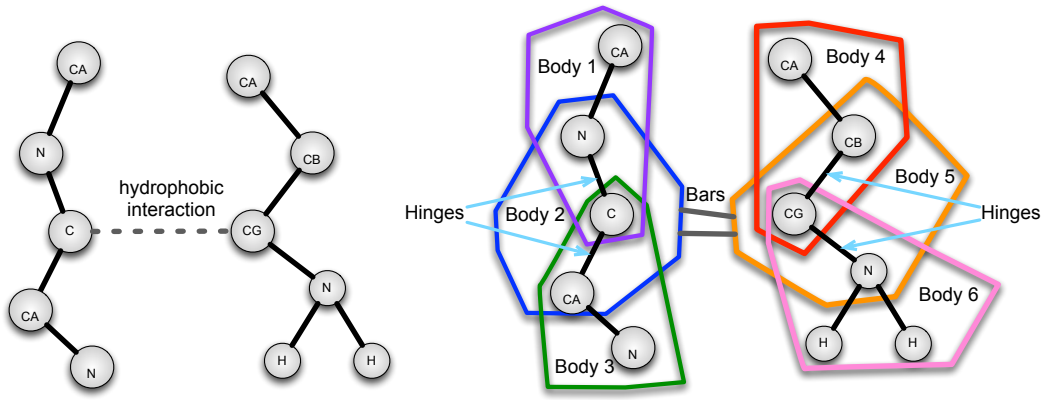
Ethane C_2H_6, shown in Figure 3.3c, is flexible. As in methane, each C atom together with its covalently-bonded neighbors, forms a rigid body comprising four atoms; since the C-C bond permits rotation, the molecule is flexible. Figure 3.3d shows the atoms of ethane clustered into two rigid bodies, forming intersecting tetrahedra, that share a rotatable bond acting as a *hinge*. The hinge model of a bond is used for rotatable covalent bonds, disulfide bonds, and strong hydrogen bonds.

Weaker interactions, such as hydrophobic interactions, can be modeled with different numbers of bars, introduced in the ASU-FIRST software [10]. The convention set in the ASU-FIRST software, and set as a default in KINARI, is to model hydrophobic interactions with 2 bars. Figure 3.4 shows an example hydrophobic interaction configuration and how KINARI would model it. The 2 bars are to be placed between the C and CG atoms, but you may notice, that each of these atoms are contained in multiple bodies. In order to build a defined Tay graph for a body-bar-hinge frame-

work, a bar endpoint must lie in one and only one body. We choose the body that contains the endpoint and all its covalent neighbors as the one to attach the bar to.

It is important to note that a multi-bar is a relaxed extension of the bar concept. A true bar, when placed between two endpoints, enforces a distance constraint. Any additional bars between the endpoints would only serve as overconstraints–no additional DOFs could be removed. These multi-bars represent the removal of DOFs, rather than specifically acting as distance constraints. The ability to increase or decrease the number of bars provides greater control to tune the system in order to achieve more rigid or flexible results. The developers of ASU-FIRST presented an interesting interpretation in order to permit the use of multi-bars in the modeling, using the concept of pseudo-atom chains [10]. Before building the constraint graph, a (conceptual-only) step is performed where a covalently-linked chain of 'pseudo-atoms' is placed between the two atoms sharing the hydrophobic interaction, as shown in Figure 3.5. This chain permits the angles and distance between the atoms to vary, but still imposes a constraint on the maximum distance. And most importantly, the chain does not introduce any degeneracies into the model. A clever observation was made that this chain need not be explicitly included in the modeling. Instead, the chain could be replaced by a multi-bar representing the number of degrees of freedom removed by the chain.

In the example in Figure 3.5, a chain of 3 pseudo-atoms connected by 4 rotatable covalent bonds has been placed. If the modeling is performed on this configuration, an additional 3 bodies and 4 hinges will be placed. Therefore, the chain permits only 4 degrees of freedom – one rotational degree of freedom for each hinge. The hinge removes 2 (6 - 4) degrees of freedom between the two bodies to which the chain is attached. The number of degrees of freedom removed can be varied by lengthening or shortening the chain.



(a) hydrophobic interaction configuration

(b) KINARI, 2 Bars

Figure 3.4: Hydrophobic interaction modeling. By default in KINARI, a hydrophobic interaction is modeled with 2 bars.

When using equivalent modeling options, there is a one-to-one correspondence between the non-overlapping bodies used in the modeling and ASU-FIRST, and those used in KINARI. But one main difference is in multi-bar modeling for hydrogen bonds. In ASU-FIRST, when the initial bodies are determined, the H and O atoms covalently bonded to only one other atom, are placed into their own rigid bodies. Figure 3.6d shows how ASU-FIRST models the hydrogen bond configuration of Figure 3.6a. This results in a (non-generic) model with 4 bodies and 3 DOFs. In KINARI, when the same hydrogen bond is modeled with 5 bars, this would results in a model with 2 bodies and 1 DOF. For backwards compatibility, we support a legacy style modeling, as shown in Figure 3.6e. Notice that there are two bodies (shows as bodies 2 and 3 in the figure) that are each completely contained in one of the two other bodies. These 'phantom' bodies are placed in order to have a one-to-one correspondence with those in ASU-FIRST.

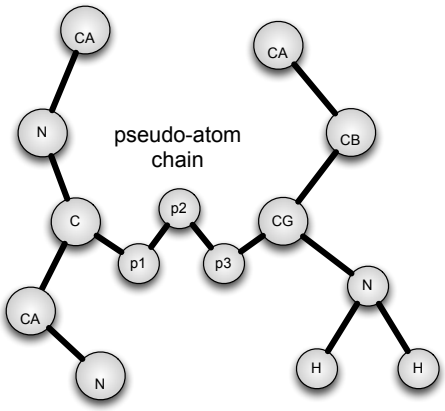


Figure 3.5: Equivalence of a pseudo-atom chain and multi-bar modeling. The chain of three pseudo-atoms, p1-p3, has 4 degrees of freedom. The constraint imposed by the chain can be incorporated into the model without including the pseudo-atoms explicitly. Instead, two bars can be placed between the corresponding bodies, equivalently to what is depicted in Figure 3.4b.

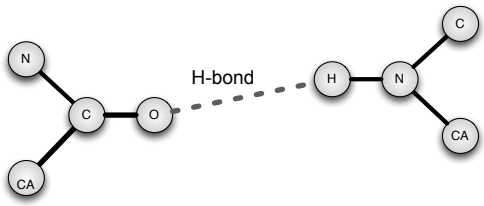
Interaction type	**How modeled**
Single covalent bond	Hinge
Double covalent bond	6 Bars
Resonance bonds	6 Bars
Disulfide bonds	Hinge
Hydrogen bond	Hinge
Hydrophobic interaction	2 Bars

Table 3.3: Default modeling parameter settings in KINARI v1.0

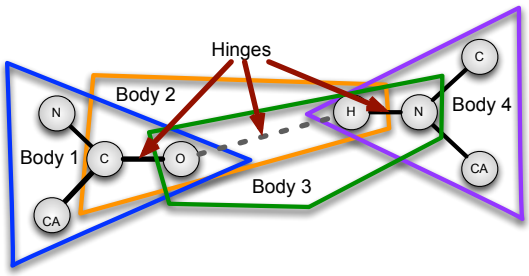
3.2.2.1 Algorithm for converting a macromolecule to a body-bar-hinge framework

In KINARI, a body-bar-hinge framework model of a molecule is built using information on the atoms and the interaction network. We describe the default converter. The available modeling options are hinge or 1 to 6 bars. We also describe *legacy-mode* for modeling hydrogen bonds.

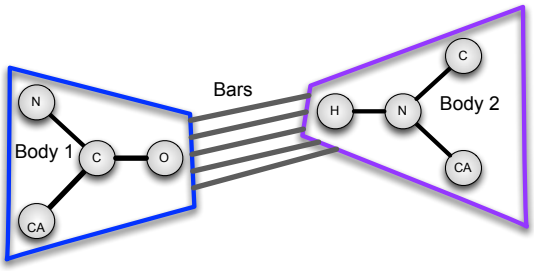
We describe the input and output objects in the conversion process, and the auxiliary data structure used during the building of the body-bar-hinge framework.



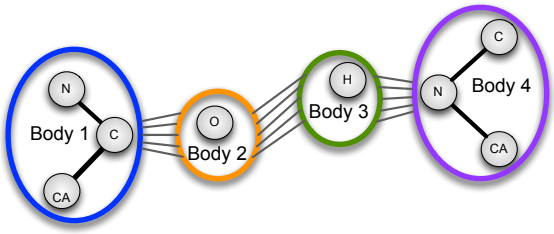
(a) H-bond configuration



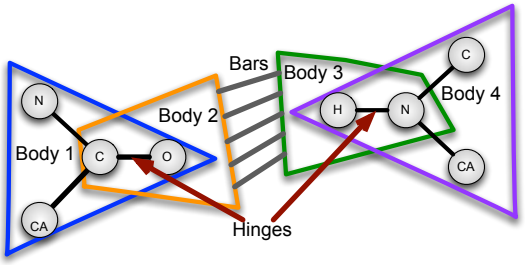
(b) KINARI, default, Hinge



(c) KINARI, 5 Bars



(d) ASU-FIRST, 5 Bars



(e) KINARI, legacy style, 5 Bars

Figure 3.6: Hydrogen bond modeling. (a) shows a hydrogen bond configuration as would be found connecting the backbone, for example in a β-sheet or α-helix. (b) The default style in KINARI is to treat a hydrogen bond equivalently as a covalent bond, modeling as a hinge, resulting in a model with 4 bodies and 3 DOFs. (c) If instead, 5 bars is specified, KINARI will place 2 bodies with 1 DOF. The number of DOFs can be increased by removing bars. (d) shows how ASU-FIRST models the configuration, resulting in 4 bodies and 3 DOFs. (e) KINARI supports a legacy mode in order to attain a model that corresponds with that of ASU-FIRST.

The described algorithm to convert a macromolecule to a body-bar-hinge framework takes linear time.

Input:

- *mol.* A macro-molecule composed of a set of atoms and a set of pair-wise interactions, including covalent bonds, hydrogen bonds, and hydrophobic interactions. These are read in from the PDB file and bond files produced by curation.
- *modeling-table.* A modeling table stores the modeling option selected for each interaction in the molecular framework. The available modeling options are HINGE, 6BARS, 5BARS, 4BARS, 3BARS, 2BARS, and 1BAR.
- *legacy-mode-flag.* A flag to specify whether to use ASU-FIRST legacy style for modeling hydrogen bonds. This only takes effect when modeling hydrogen bonds with 1 to 6 bars. This modeling option simply specifies whether additional bodies be placed for H-bonds. The first extra body consists of the H and the donor atom, and the second consists of the C=O for backbone hydrogen bonds.

Output:

- *bbh.* A body-bar-hinge framework model of the molecule consisting of:
  - A set bodies. Each body consists of a set of atoms.
  - A set of bars. Each bar connects exactly two bodies at two atom endpoints.
  - A set of hinges. Each hinge connect exactly two bodies at two atom endpoints. The two atoms must belong to each body of the hinge.

Auxiliary data structure:

- *cmap.* A central-atom-to-body map tracks the central atom for each body. Each body consists of a central atom and its 'hinge' neighbors, atoms that share an interaction with the central atom that is modeled as a hinge in the modeling

table. A 'bar' neighbor is an atom connected with an interaction modeled by 1BAR to 6BARS. These are not included in the body.

Algorithms 1 contains the pseudocode for converting a molecular object into a body-bar-hinge framework. It uses the functions listed in Algorithms 2, 3, and 4 as subroutines. Algorithm 4, BuildBarsAndHinges, uses Algorithm 5, GetBody, as a subroutine. GetBody queries *cmap* or *bbh* for efficiently determining the bodies to connect with bar and hinge constraints.

Algorithm 1 ModelMoleculeAsBodyBarHinge()

```
InitializeBBHModeler()
BuildBodies()
BuildBarsAndHinges()
```

Algorithm 2 InitializeBBHModeler()

```
Initialize bbh as an empty body-bar-hinge framework
Initialize cmap to empty
```

Algorithm 3 BuildBodies()

```
for all atoms, a, in mol do
    Create a new body, b, and add a
    Collect all 'hinge' neighbors of a and add to b
    if b has 3 or more atoms or a is in a hydrogen bond and legacy-mode is True
then
        add b to bbh
        update cmap, cmap[a] = b
    else
        discard b
    end if
end for
```

3.2.2.2 Converting to residue-level clusters

KINARI employs an all-atom model to determine rigid clusters of atoms. For comparison with other methods (as we will do in Chapters 4 and 5), it is oftentimes necessary to have data on the rigidity of the backbone only. In order to present the

Algorithm 4 BuildBarsAndHinges()

for all interactions i in mol, between atoms $a1$ and $a2$ **do**
  $b1 = GetBody(a1)$
  $b2 = GetBody(a2)$
  look up modeling for i in $modeltable$
  if i is modeled as a $HINGE$ **then**
    add new hinge (a1, a2, b1, b2) to bbh
  else if i is modeled as $1BAR$, $2BARS$, ..., $6BARS$ **then**
    add new $bar(a1, a2, b1, b2)$ with multiplicity of 1, 2, ..., 6
  end if
end for

Algorithm 5 GetBody(a)

if a is in $cmap$ **then**
  return $cmap[a]$
else
  return the only body in bbh that contains a
end if

information in a residue-level format, we examine the body-bar-hinge model output by KINARI. For each atom-level rigid cluster determined by KINARI, we collect the residues whose CA atoms belong to the cluster. For each such CA atom, we examine the C-CA and CA-N bonds. If neither corresponds to a hinge in the body-bar-hinge model, meaning that the rotation is inhibited by the network of chemical constraints, we add the CA atom's residue to the residue cluster.

3.3 System description

Figure 3.7 shows the different software components that fall under the KINARI project, a collaborative efforts in Streinu's Linkage Laboratory. Ileana Streinu is the project lead and the main designer. In particular, she came up with the novel modeling scheme. Audrey Lee-St John consulted in early stages of design. The system was built after discussions with Michael F. Thorpe, one of the pioneers of protein rigidity analysis, and Brandon Hespenheide, who developed the FIRST systems in his PhD thesis work and later as a researcher in Thorpe's lab. They shared the ASU-

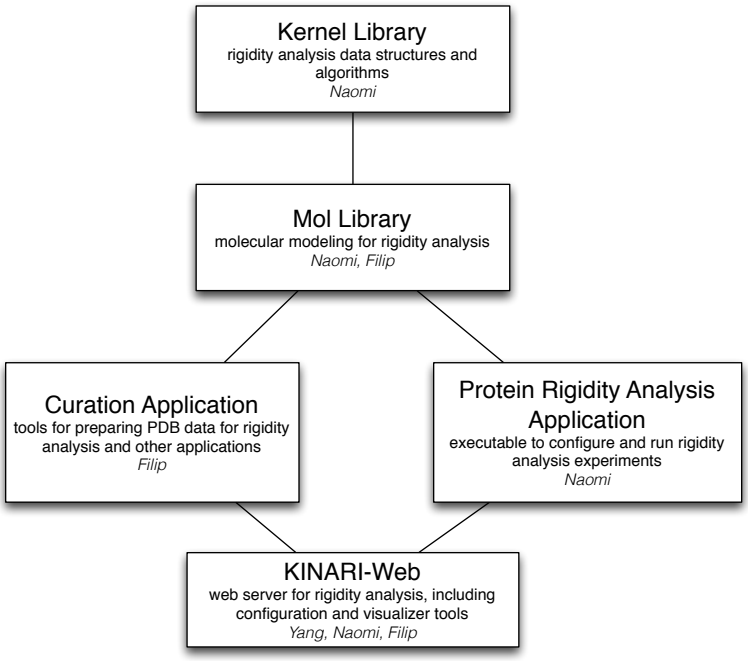


Figure 3.7: KINARI Components (duplicate of Figure 1.2)

FIRST code base, as well as important implementation details early on, in a visit I made to Thorpe's Biophysics lab at Arizona State University in 2007.

In my thesis work, I implemented the kernel library and modeling code in the molecular library. Filip Jagodzinski implemented the non-trivial task of PDB file parsing, and performed much of the profiling. His thesis work focuses on correlating rigidity metrics with protein stability data derived from laboratory experiments. Yang Li, Smith 2011, wrote the bulk of the code for the web front-end and Jmol-based visualizer.

We describe some important details of the system design. The first released version of KINARI-Web, with its default settings, is referred to through the rest of this thesis as KINARI v1.0. The curation and modeling settings for KINARI v1.0 are listed in Tables 3.2 and 3.3.

3.3.1 Kernel Library

The kernel library, released under the name KINARI-Lib, has classes for pebble games and for 3D body-bar-hinge framework mechanical modeling [22]. To support this functionality, the kernel library also provides classes for representing graphs, calculating some statistics on the body-bar-hinge frameworks, and for reading and writing each data structure in XML. A brief description of the key classes is included below and in Figure 3.9. Complete API documentation is distributed with the software, and the tutorial is available for download from the KINARI website.

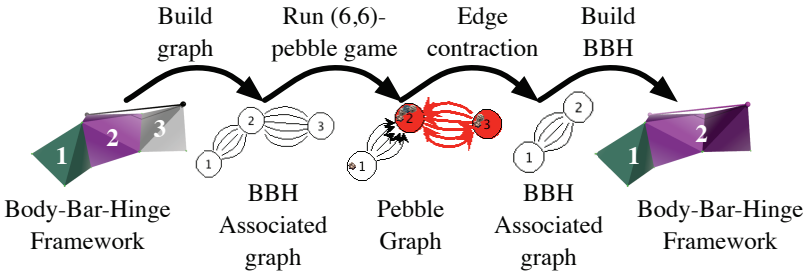
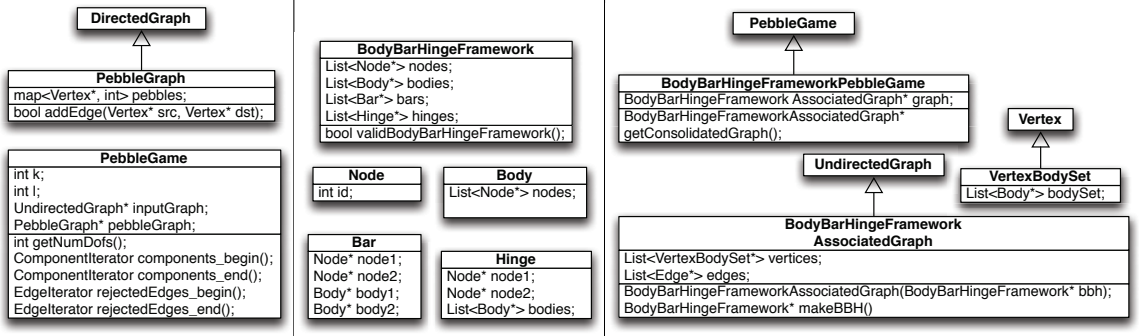


Figure 3.8: Rigidity analysis applied to a generic body-bar-hinge framework.

`PebbleGame.` This class contains an implementation of the component pebble game, described in [64]. The constructor runs it by default, and requires the (k, ℓ) sparsity parameters and a (multi)graph as input. Functions are provided to retrieve the number of degrees of freedom, the maximal (rigid) components, and to identify over-constrained edges. The class contains one `PebbleGraph` object as a member variable. A `PebbleGraph` is a special extension of a directed multi-graph, on which the pebble game is played.

`BodyBarHingeFramework.` This class contains lists of nodes, bodies, bars, and hinges. The nodes are a discrete set of points used to define the bodies, bars, and hinges. The `Node` class represents a point, that can (optionally) have coordinates associated with it. The `Body` class contains a set of nodes. The `Bar` represents the bar connection between two Body objects at two Node endpoints. A



(a) (k, ℓ)-Pebble Game (b) Body Bar Hinge Framework (c) Body Bar Hinge Framework Pebble Game

Figure 3.9: UML class diagram of selected classes from KINARI Kernel library (KINARI-Lib).

`Hinge` connects at least two bodies, at two nodes defining the hinge axis. The `BodyBarHingeFramework::validBodyBarHingeFramework()` function can be called to check a number of conditions that are described in the API documentation.

`BodyBarHingeFrameworkPebbleGame.` This class provides special functionality applicable to the pebble game on body-bar-hinge frameworks. The input to this is an object of the `BBHFwkAssociatedGraph` class. The `BBHFwkAssociatedGraph` a special undirected graph associated with a body-bar-hinge framework. Each vertex points to one (or more) bodies, and the edges have bars or hinges associated with them. In this way, the post-pebble-game body-bar-hinge framework can be built. The `BodyBarHingeFrameworkPebbleGame::getConsolidatedGraph()` function is provided to output an edge-contracted graph, where each component has been consolidated into a single vertex. Because this graph is of type `BBHFwkAssociatedGraph`, it contains all the information necessary to build the simplified `BodyBarHingeFramework`, where each rigid body is maximal.

We include two short examples illustrating how the kernel libary functionality can be incorporated into a C++ program. The first example runs the pebble game on

```
#include <PebbleGame.h>
#include <GraphXMLFileIO.h>
#include <ComponentsXMLWriter.h>
using namespace Kinari;
int main( int argc, char **argv ) {
  // 4 command line parameters:
  // 1. The name of the XML file containing the input graph
  std::string graphfilename(argv[1]);
  // 2. k in the (k,l) values required to configure the pebble game
  int k = atoi(argv[2]);
   // 3. l in the (k,l) values required to configure the pebble game
  int l = atoi(argv[3]);
   // 4. The output file for the components
  std::string componentfilename(argv[4]);
  try {
    GraphXMLFileIO graphReader(graphfilename);
    UndirectedGraph* ugraph = graphReader.extractGraph();
    PebbleGame pg(k,l, ugraph);
    ComponentsXMLWriter::writePebGameResultsFile(pg, componentfilename);
  } catch (KinariException e) {
      std::cerr << e.toString() << std::endl; }}
```

Figure 3.10: Example code for invoking pebble game.

graphs. The second one analyzes the rigidity of 3D body-bar-hinge frameworks. The library distribution includes extended versions of the two examples.

General graph pebble game. This example code, shown in Figure 3.10, performs the following steps:

1. Read a graph from file.
2. Run the pebble game on the graph, with user-specified (k, ℓ) values
3. Write an XML file containing the components, DOFs, and over-constraints calculated by the pebble game.

The error handling utilities inside the KINARI code are also demonstrated. Here, the `PebbleGame` constructor will throw a `KinariException` if the (k, ℓ) values do not fall in the acceptable range.

Rigidity analysis of Body-Bar-Hinge Frameworks. The commented code example in Figure 3.11 shows the KINARI classes and syntax for analyzing a body-bar-

```
#include <BBHFwkXMLFileIO.h>
#include <GraphXMLFileIO.h>
#include <BodyBarHingeFrameworkPebbleGame.h>
using namespace Kinari;
int main( int argc, char **argv ) {
  // Read a body-bar-hinge framework from an XML file
  BBHFwkXMLFileIO bbhXMLReader("bbh.xml");
  BodyBarHingeFramework* bbh = bbhXMLReader.extractBBH();
  // Create an associated graph
  BBHFrameworkAssociatedGraph bodyGraph(bbh);
  //Play pebble game and get the output graph with edge contractions
  BodyBarHingeFrameworkPebbleGame pg(&bodyGraph);
  BBHFrameworkAssociatedGraph* edgeContractedBBHGraph = pg.getConsolidatedGraph();
  GraphXMLFileIO::writeOutGraph(*edgeContractedBBHGraph,
"edgeContractedBBHGraph.xml");
  // Retrieve minimized body-bar-hinge and write to file
  BodyBarHingeFramework* consolidatedBBH = edgeContractedBBHGraph->makeBBH();
  BBHFwkXMLFileIO::writeXMLToFile(*consolidatedBBH,"postPG_BBH.xml");}
```

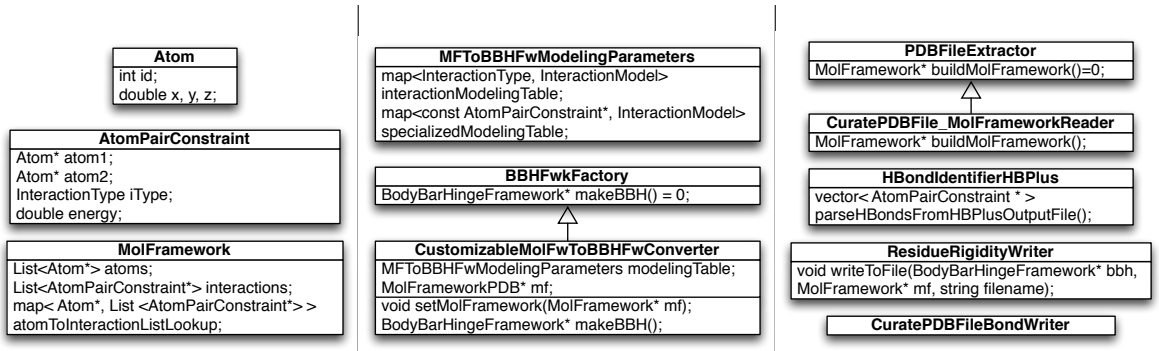
Figure 3.11: Example code for body-bar-hinge framework rigidity analysis.

hinge framework using the BodyBarHingeFrameworkPebbleGame class. The steps performed mirror those shown in Figure 3.8.

3.3.2 KINARI Molecular library

The KINARI molecular library contains all classes to facilitate molecular modeling for rigidity analysis.

Figure 3.12 shows a UML-style diagram of simplified versions of some of the important classes. The `MolFramework` class holds all the molecular data and supports fast queries of the atom-interaction network. The most important classes in the library involve conversion between a molecule to a body-bar-hinge object. The `CustomizableMolFwToBBHFwConverter` contains a reference to a modeling customization table, of type `MFToBBHFwModelingParameters`. This class supports default modeling for each interaction type, and if desired, special modeling for particular interactions. This specialized modeling feature is used for the energy-refined modeling features described in Chapter 5 of this thesis. File I/O classes provide support for parsing data from PDB files and identifying relevant bonds and interactions.



(a) Data Structures (b) Converters (c) File I/O

Figure 3.12: UML class diagram of selected classes from KINARI Molecular library.

The main class for parsing PDB file input, calculating interactions, and building a MolFramework is the `CuratePDBFileMolFrameworkReader` class. Although most interactions are calculated internally by KINARI, we do rely on the HBPLUS software to calculate hydrogen bonds [69]. The `HBondIdentifierHBPlus` class supports parsing HBPLUS output files.

The code example of Figure 3.13 shows how the KINARI Molecular library classes are used to:

1. Read in a molecular object from a PDB file and interactions files that are output by curation.

2. Build a body-bar-hinge framework model of the molecular object.

3. Perform rigidity analysis on that molecule and output the results in an XML-format file.

3.3.3 KINARI-Web

KINARI-Web is a publicly-available web server for protein rigidity analysis (`http://kinari.cs.umass.edu`). KINARI-Web was developed by Jagodzinski, Li, Streinu, and myself. My main contribution in this endeavor was implementing the back-end

```
#include <CuratePDBFile_MolFrameworkReader.h>
#include <CustomizableMolFwToBBHFwConverter.h>
#include <BodyBarHingeFrameworkRigidityAnalyzer.h>
#include <BBHFwkXMLFileIO.h>
#include <string>
#include <iostream>
using namespace std;
using namespace Kinari;
// required command line arguments:
// 1. pdb file name
// 2-7: interactions file names
// 8: body-bar-hinge framework output file name
int main(int argc, char **argv) {
        // Step 1: initialize and configure MolFramework builder
        CuratePDBFile_MolFrameworkReader reader(argv[1]);
        reader.setBondFileName(SINGLECOVBOND, argv[2]);
        reader.setBondFileName(DOUBLECOVBOND, argv[3]);
        reader.setBondFileName(RBOND, argv[4]);
        reader.setBondFileName(HBOND, argv[5]);
        reader.setBondFileName(HYDROPHOBICTETHER, argv[6]);
        // Step 2: Build your molFramework object and retrieve
        reader.buildMolFramework();
        MolFrameworkPDB* molFramework =
        static_cast<MolFrameworkPDB*>(reader.getMolFramework());
        // Step 3: model the MolFramework as the initial BodyBarHingeFramework
        CustomizableMolFwToBBHFwConverter modeler(molFramework);
        BodyBarHingeFramework* initialBbh = modeler.makeBBH();
        // Step 4: Run the pebble game rigidity analysis
        BodyBarHingeFrameworkRigidityAnalyzer rigidityAnalyzer(initialBbh);
        BodyBarHingeFramework* minimizedBbh = rigidityAnalyzer.makeBBH();
        // Step 5: Write the post pebble game bbh to file
        BBHFwkXMLFileIO::writeXMLToFile(*minimizedBbh, argv[7]);
}
```

Figure 3.13: Example code for protein rigidity analysis.

protein rigidity analyzer executable. This was configurable with an XML format input file. This configuration file lists the input PDB and interactions file, as well as any custom modeling options. We were very fortunate to work with Smith undergraduate Yang Li, who implemented the bulk of the PHP code for the server. Having access to this code base greatly eased the implementation of extensions, namely the Redundancy analyzer described in the next section. I also contributed substantially to the front-end design, testing, and debugging.

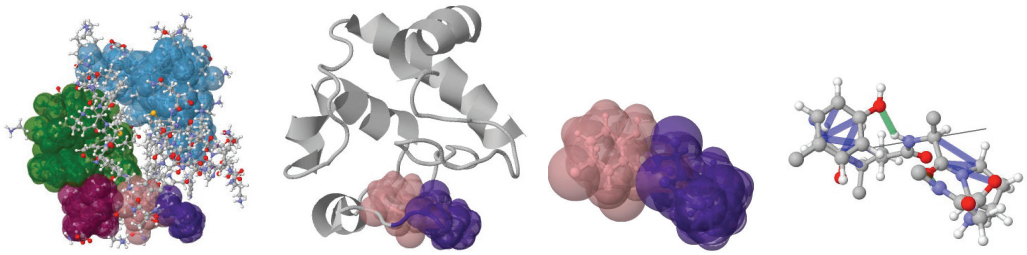
3.3.3.1 Features

KINARI-Web enhances the KINARI command-line applications with a web-based front-end. It provides tools for streamlining the curation of the input protein data file and for building a molecular model that can be customized by the user. A record of the performed experiments is provided in text files containing all options needed to reproduce the results. For beginners, KINARI-Web offers a quick-start alternative that sets curation and modeling parameters to default values.

A key feature is the interactive Jmol-based visualization tool for exploring the rigidity results. Each cluster is highlighted with a different color. Because two clusters can overlap, the clusters are shown with colored surfaces rather than by coloring each atom a single color. By default only the large rigid bodies are shown. The system provides a list of all the calculated clusters, from which the user can select which ones to display and which ones to hide; this makes it possible to view rigid bodies in isolation or in context. The user can zoom in to investigate specific regions, such as known active sites or domain interfaces that are functionally or structurally important. Bonds that act as hinges between rigid clusters, surfaces for each rigid cluster, specific atoms, and different chemical interactions can be displayed or hidden from view. The ability to view and investigate the rigidity properties of specific small regions of a protein is a visualization feature that is not available elsewhere.

Other notable features of the visualizer include:

- Display hydrogen bonds and hydrophobic interactions.
- Show all atoms in ball-and-stick or cartoon mode. Shapes of rigid clusters are shown with highlighted surfaces.
- Select which clusters to show, filtering by size, or select cluster IDs from drop-down menu.
- Hide or show atoms to help better visualize regions of interest.
- Return to the ball-and-stick default view (shows the rigid clusters with at least 20 atoms each) at any time.
- Full functionality of *Jmol*: zoom-in and out, translate protein, and click on atoms to get their names and IDs.
- Save an orientation, that can be reverted back to.
- Take a snapshot and save as a jpeg file for later use.
- *Hinge and bar options:* display all hinge axes and hinge bonds; select a particular hinge from a drop-down menu; show the two clusters of the hinge in isolation (hiding all other atoms), optionally highlight the two clusters with surfaces; apply spin. This will rotate the molecules about selected hinge axis.



(a) Rigidity results with default options (b) Viewing specific clusters (c) Viewing hinge regions (d) Hinge between 2 bodies

Figure 3.14: Demonstration of the KINARI-Web interactive visualizer. The visualizer can be used to display different rigidity features of 1HRC. The quick-start option uses default curation and modeling options, and displays the largest rigid clusters as highlighted surfaces (a) in a bar-and-stick model. Many other visualization options are available. The same protein can be viewed as a cartoon (b), two bodies that connect at a mechanical hinge can be shown (c), while hydrogen bonds and hydrophobic interactions can be displayed in the vicinity of a mechanical hinge region.

3.3.3.2 Case study of Cytochrome-*c* (1HRC), to demonstrate KINARI-Web

Included here is a case study to demonstrate the visualization features. Other case studies using KINARI-Web are available from the KINARI website.

Horse heart Cytochrome-*c* is a 105 residue heme protein associated with the inner membrane of mitochondria. The rigidity of this protein was previously investigated by [40,90,94]. To analyze it with KINARI, we invoked the quick-start analysis option with PDB code 1HRC. The curating of the PDB file (the ligand was automatically removed) and the rigidity analysis using default options were performed in less than 5 seconds.

The calculated rigid regions of a protein can be easily explored in the visualizer. Figure 3.14a shows the ball-and-stick model rendition of 1HRC in the *Jmol* applet embedded in KINARI-Web. The 5 rigid bodies that are composed of at least 20 atoms are displayed with randomly-colored surfaces. Different visualizer features can be used to customize the parts of the protein and the set of rigid clusters displayed. Figure 3.14b shows the same protein, but the cartoon display option is selected, and only a subset of the clusters are highlighted.

The pink and purple clusters share a rotatable bond that acts as a hinge. The two clusters can be displayed in isolation, with highlighted clusters as in Figure 3.14c, or, as in Figure 3.14d, with the hinge axis, hydrogen bonds (green) and hydrophobic interactions (blue). The hydrogen bonds and hydrophobic interactions hold the two clusters rigidly together, but do not cross-link between the clusters. Visualizing the hinge up-close gives insight into the range of motion that the bond might exhibit.

3.3.4 KINARI-Redundancy

Chapter 6 contains a description of the KINARI-Redundancy web application. This application supports all of the curation and modeling options of KINARI-Web

Protein	PDB	MSU-FIRST	KINARI v1.0
LAO-binding (closed)	1LST	1	1
LAO-binding (open)	2LAO	1	1
HIV-1 Protease (closed)	1HTG	4	3
HIV-1 Protease (open)	1HHP	3	3
Dihydrofolate Reductase (closed)	1RX1	2	1
Dihydrofolate Reductase (occluded)	1RX6	2	2
Dihydrofolate Reductase (open)	1RA1	2	2
Adenylate Kinase (closed)	1AKY	6	6
Adenylate Kinase (open)	1DVR	4	3

Table 3.4: A comparison of the flexible loop regions detected by MSU-FIRST and KINARI v1.0. The loops are annotated in [48].

standard rigidity analyzer. But on top of that, it provides the user with access to the critical and redundant interactions found within the rigid clusters.

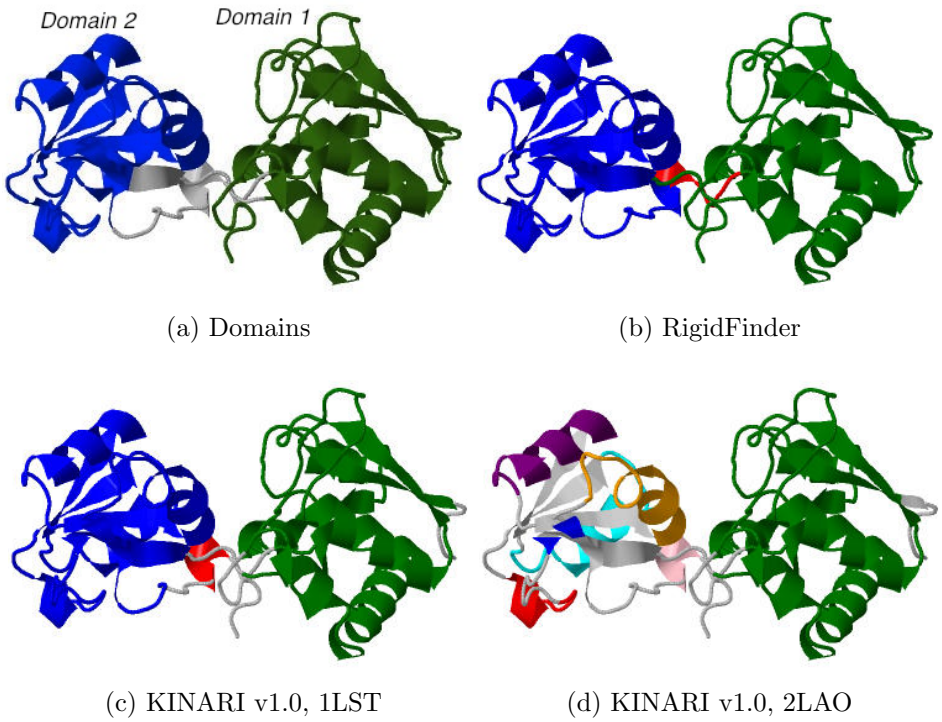
3.4 Case studies comparing results of KINARI v1.0 with previously published rigidity analysis results

We describe in detail case studies comparing results from KINARI v1.0 (see Tables 3.2 and 3.3), and those reported for MSU-FIRST [47,48]. We found the KINARI v1.0 decompositions matched well with the MSU-FIRST decompositions, but there were some subtle differences. We also include a comparison with results produced by the RigidFinder method, described earlier in Chapter 2, Section 2.3.2.

Overall, the cluster decompositions produced by the two methods, visually, had high overlap in the rigid clusters and flexible regions identified. The MSU-FIRST report confirms literature annotated flexible loops with those identified by the software. For most of the cases, KINARI identifies the same flexible loops as does MSU-FIRST. Table 3.4 summarizes the counts of loops detected by MSU-FIRST and matched by KINARI.

3.4.1 Case study of Lysine-Arginine-Ornithine Binding Protein

The lysine-arginine-ornithine binding protein (LAO, which transports important substrates in bacteria, has a bi-lobal or"clam-shell", structure. The two LAO crystal structures used in the original MSU-FIRST study were an open (2LAO) and closed (1LST) structures [47]. It is composed of two stable domains: domain 1 (residues 1-87, 195-237, containing N- and C- terminals) and domain 2 (residues 94-181). The remaining region consists of loops forming a domain-level hinge. See Figure 3.15a.



(a) Domains (b) RigidFinder

(c) KINARI v1.0, 1LST (d) KINARI v1.0, 2LAO

Figure 3.15: Rigid cluster decompositions of lysine-arginine-ornithine binding protein. All of the decompositions are depicted on 2lao.

The MSU-FIRST results report that the residues of domain 1 lie in a single rigid cluster. It was observed that domain 2 is more flexible. Smaller rigid cluster, mainly composed of α-helices, form within the domain, but the β-sheet remains flexible. There are slight differences in the distribution of the rigid clusters between the open and closed conformations, but for both conformations, the MSU-FIRST software

predicts domain 2 to be rather flexible. The main difference between the two decompositions was that for the open conformation, the flexible domain-level hinge region is larger, extending further into domain 2 than for the closed conformation. The open conformation is expected to be more flexible than the closed conformation because the interfaces are separated and there are fewer opportunities for hydrogen bonding and hydrophobic interactions.

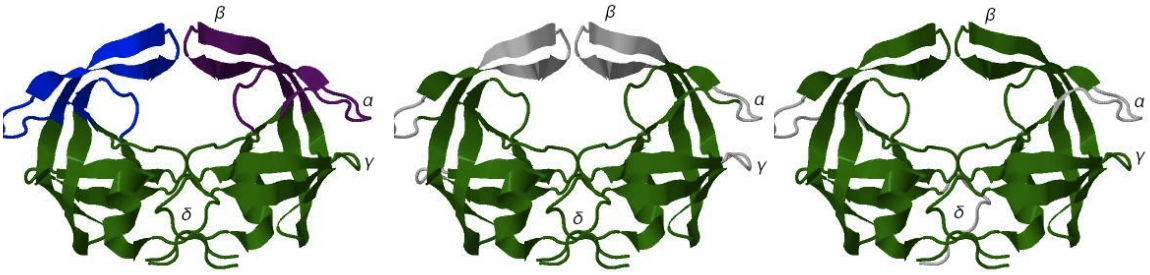
In the KINARI decompositions, the difference in flexibility between the open and closed conformations is more stark. The KINARI decomposition on the closed conformation (2LAO, Figure 3.15d) reflects the same level of flexibility as that produced by MSU-FIRST. Like the MSU-FIRST decomposition, domain 1 lies in a single rigid cluster while domain 2 is composed of smaller, mainly α-helix, rigid clusters and a flexible β-sheet. The decomposition for the closed conformation (1LST, Figure 3.15c) does not show the same flexibility in the 2nd domain. The two domains are both placed into their own rigid clusters.

It appears that the differences between the systems, mainly the inclusion of the hydrophobics and the underlying rigidity analysis modeling and algorithm, have had subtle but important differences for the cluster decompositions between the open and closed structures.

For another comparison, we used RigidFinder [1] to decompose the protein into rigid domains, based on the open and closed conformations. See Figure 3.15b). RigidFinder decomposes the protein into exactly three domains: one that matches domain 1, another that matches domain 2, and a third composed of the domain-level hinge region. The RigidFinder decomposition does not identify the flexibility within domain 2.

3.4.2 Case study of HIV-1 Protease

For the open and closed forms of HIV-1 Protease (1HHP, 1HTG), the results reported by MSU-FIRST and KINARI have good correspondence. For the closed conformation, the KINARI residue-level cluster decomposition was the same whether the ligand was present or removed from the analysis. For 1HHP, both MSU-FIRST and KINARI identify the single, dominating rigid cluster and the three flexible regions (labeled as α, β, and γ), Figure 3.16b. The large rigid cluster contains the base and walls of the binding cavity. For the closed form (1HTG), KINARI and MSU-FIRST results both reflect the increase in rigidity upon binding. The large rigid cluster now includes the α and β regions, but not the γ region.



(a) RigidFinder (b) KINARI v1.0, 1HHP (c) KINARI v1.0, 1HTG

Figure 3.16: Rigid cluster decompositions of HIV-1 Protease. All of the decompositions are depicted on the 1HHP dimer.

One interesting difference is in the δ region is the dimer interface, composed of residues at the N- and C-termini that do not belong to a secondary structure. Although the KINARI results show flexibility in the δ in chain A, chain B has maintained rigidity. In the MSU-FIRST decomposition, the entire δ region is flexible. The loops above the δ region is the catalytic site, containing the characteristic Asp-Thr-Gly sequence (Asp25, Thr26 and Gly27) common to aspartic proteases, lie within the rigid core for both the open and closed conformations (computed both by KINARI and ASU-FIRST). Functionally, the δ region is where the two monomers are held together; none of the many known drug resistance mutation sites are located within

δ [93]. The importance of the findings on the different levels of flexibility within the regions is unclear.

The RigidFinder decomposition for HIV-1 Protease has some interesting differences from the either of the KINARI or MSU-FIRST decompositions. The α and γ regions, the β-sheet that both regions share, as well as loops forming the wall of the binding cavity, have been placed into a single cluster. The rest of the protein, including γ and δ regions compose a second rigid cluster.

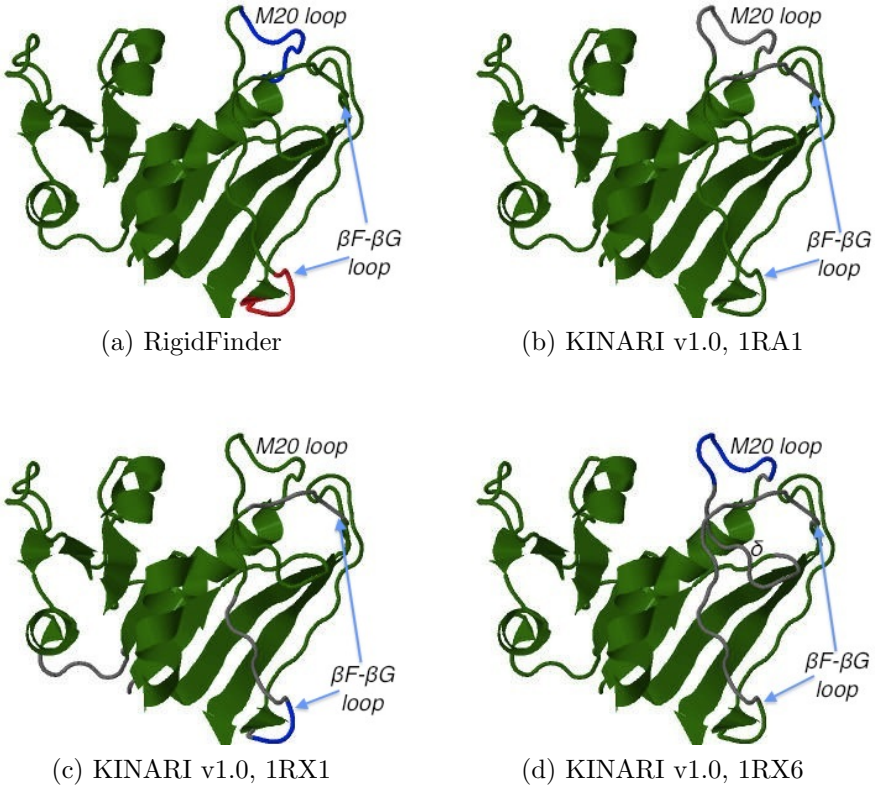
3.4.3 Case study of Dihydrofolate Reductase

We compare the MSU-FIRST decompositions of 3 conformational states of dihydrofolate reductase (open, 1RA1; closed, 1RX1; occluded, 1RX5) with those from KINARI with default options. The KINARI decompositions were in general much more rigid than those reported by MSU-FIRST. The labeled M20 and βF-βG loops are of key importance to binding specificity and should be flexible. Overall, the KINARI results report a larger dominating rigid cluster. Even with the exclusion of the ligand from the analysis, the entire protein, other than the 2-3 flexible loops reported, is included in a single rigid cluster. We now compare in further detail the results reported by MSU-FIRST and KINARI in detecting flexibility in the loop regions.

For 1RA1 (Figure 3.17b), MSU-FIRST correctly detects flexibility in the M20 loop, a captures the flexibility of a subsection of the βF-βG loop. KINARI detects the same regions of flexibility as MSU-FIRST on 1RA1.

For 1RX1 (Figure 3.17c), part of the M20 loop is detected by MSU-FIRST to be flexible, while most of the βF-βG loop is detected to be flexible. KINARI does not detect any flexibility in the M20 loop, but detects the same flexible region as MSU-FIRST in the βF-βG loop.

The MSU-FIRST results for 1RX6 (Figure 3.17d) are similar to those it reports for 1RX1, but an even larger region of flexibility is detected in the βF-βG loop. The



(a) RigidFinder (b) KINARI v1.0, 1RA1

(c) KINARI v1.0, 1RX1 (d) KINARI v1.0, 1RX6

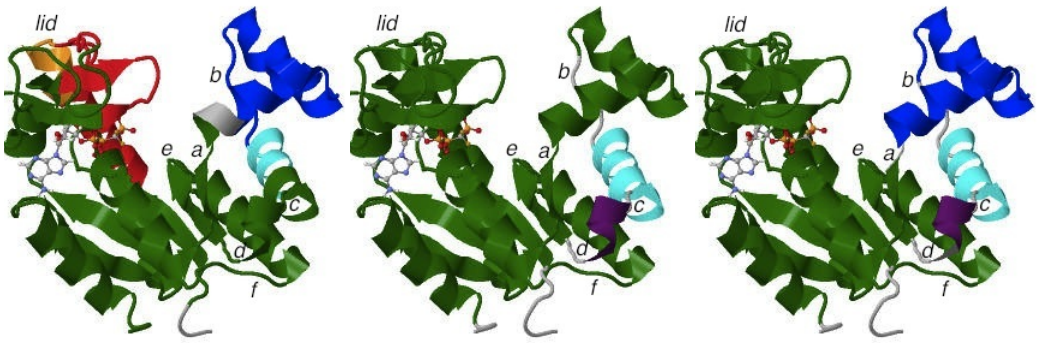
Figure 3.17: Rigid cluster decompositios of Dihydrofolate Reductase. All of the decompositions are depicted on 1RA1.

flexible region of the M20 loop is more extensive (labeled δ) and the entire βF-βG loop is detected to be flexible.

The RigidFinder decomposition (Figure 3.17a), based on 1RA1 and 1RX1, places most of the protein into a single domain, except for a piece of the M20 loop and a small segment of loop adjacent to the βF-βG loop.

3.4.4 Case study of Adenylate Kinase

Adenylate kinase undergoes a domain-level hinge motion upon ligand binding with a 2-step mechanism. Figure 3.18 shows decompositions of adenylate kinase, depicted on the ATP-bound, open conformation (1DVR). The domain containing the binding



(a) RigidFinder (b) KINARI v1.0, 1AKY (c) KINARI v1.0, 1DVR

Figure 3.18: Rigid cluster decompositios of Adenylate Kinase. All of the decompositions are depicted on 1DVR.

domain is labeled as the lid-domain. The AP_5A-bound, fully-closed conformation (1AKY, not shown) was analyzed.

For 1AKY, MSU-FIRST and KINARI decompositions match quite closely, with a nice alignment between the clusters and flexible regions between the two decompositions. However for the open conformation (1DVR), there are some subtle differences between the two decompositions. MSU-FIRST identifies 6 flexible loop regions (labeled a-e). Of these, KINARI's flexible regions match with all but 2 of them. Instead, KINARI includes these two loop regions (labeled as e and f) in rigid clusters. KINARI identifies the entire lid region as rigid, while in MSU-FIRST, the loops on the tip of the region have been identified as flexible.

3.5 Conclusion

We have designed and built KINARI to serve as a general library to support mechanical modeling. In this chapter, we provided details on the curation and modeling concepts important for reproducibility of our work, including the default parameter settings for KINARI v1.0.

The first application of KINARI is to protein modeling. The kernel library (publicly released as KINARI-Lib) has been designed to readily integrate into other applications that may benefit from rigidity analysis, such as computer-aided design (CAD) [37] or sensor network localization [63].

CHAPTER 4

BENCHMARKING A RIGIDITY ANALYSIS SYSTEM

Our main goal in developing KINARI is to validate the predictive power of protein rigidity analysis. Rigidity analysis, as a tool for determining rigid cluster decompositions (RCDs), has been validated on only a handful of proteins [1].

In this chapter, we will propose a method that goes beyond a case study-based validation. We use a benchmarking data set and score the accuracy of the RCDs determined with KINARI v1.0. We find that predictive power of KINARI v1.0 is significant for the larger ($>$ 500 residues) proteins, but performs worse than the all-floppy or all-rigid baselines for the medium- and small-sized proteins. Our results highlight the need to develop new methods which more accurately capture the rigidity and flexibility properties of all proteins.

4.1 Introduction

As new generations of bioinformatics systems are released with new features and updated methods, it is vital to ensure that their results continue to match or improve upon previous generations. A number of protein rigidity analysis software systems have been built, including MSU-FIRST (now ProFlex) [48], ASU-FIRST [10], and our own KINARI [21]. All of these take as input a single protein structure in a PDB file and output a decomposition of the protein into rigid clusters. Although all

[1]The dilution extension of rigidity analysis was more extensively validated as a method for computationally determining the folding core. This is different task from accurately determining the rigidity and flexibility at the native state.

the systems share the same general approach of mechanical modeling and running a pebble game algorithm, there are substantial differences in both their modeling and in the underlying algorithms.

Previously, case studies were used to validate the rigid cluster decompositions produced by rigidity analysis against data from laboratory experiments [48]. Providing case analyses for validation should not be under-appreciated as a contribution. Case studies demonstrate how the system may be used to test real hypotheses on protein flexibility. Yet, they do not facilitate measurement of improvements. If a new case study does not provide the desired results, one may spend months tuning curation and modeling parameters until these are attained. A quantitative approach to evaluation compliments these case studies, by providing a fast way to measure improvement.

Contribution. We propose a general methodology for benchmarking protein rigidity analysis systems. Included in this a method to assign a score to a predicted cluster decomposition, compared with decompositions produced by some other method. This is an adaptation of the B-cubed score from the information retrieval literature, which is used as a comparative score on two clusterings of the same data [2]. We use this evaluation method to benchmark our software, KINARI. In our benchmarking we use two data sets: the first is composed of several proteins used to validate the MSU-FIRST software [47, 48] and the second is used in the Gerstein Lab to validate the RigidFinder server [1]. We have made the benchmarking scripts, written in Python, available at the KINARI web site for public use.

4.2 Methods

In the next sections, we discuss the B-cubed scoring method, the data set used in our evaluation, and the scripts developed for performing the evaluation.

4.2.1 Comparative cluster decomposition scoring

To evaluate our new methods for including and modeling hydrogen bonds and hydrophobic interactions, we propose the application of a scoring method from the information retrieval literature, called the B-cubed scoring method [2]. The score is a measurement of the similarity of two clusterings of the same data. We will use it to compare cluster decompositions produced by KINARI (the model) with a curated set of *gold standard* decompositions. For each 'item' (data point, document, residue, etc), the *precision* is the fraction of items in its predicted cluster that also lie in its cluster in the gold standard. The *recall* is the fraction of items in its gold standard which are also in its predicted cluster. The F1-score combines the precision and recall into one score.

We more formally define the concepts now. For each item i, $GS(i)$ is the cluster it belongs to in the gold standard decomposition. Similarly, $M(i)$ is i's cluster in the model's predicted decomposition. $Pr(i)$ and $Re(i)$ are, respectively, i's precision and recall. The precision and recall of a decomposition D, $Pr(D)$ and $Re(D)$, are simply the mean precision and recall of the items. $F1(D)$, the F1-score of D, is the harmonic mean of $Pr(D)$ and $Re(D)$. The following five equations show how $Pr(i)$, $Re(i)$, $Pr(D)$, $Re(D)$, and $F1(D)$ are calculated.

$$Pr(i) = \frac{|GS(i) \cap M(i)|}{|M(i)|} \tag{4.1}$$

$$Re(i) = \frac{|GS(i) \cap M(i)|}{|GS(i)|} \tag{4.2}$$

$$Pr(D) = \frac{1}{n} \sum_{i=1}^{n} Pr(i) \tag{4.3}$$

$$Re(D) = \frac{1}{n} \sum_{i=1}^{n} Re(i) \tag{4.4}$$

$$F1(D) = \frac{2 * Pr(D) * Re(D)}{Pr(D) + Re(D)} \tag{4.5}$$

All-floppy and all-rigid baselines. For a set of items, the two most extreme ways of naively decomposing are a completely floppy prediction (placing each item into its own unique cluster) or a completely rigid prediction (placing all items into the same cluster). These two methods result in 100% precision and 100% recall, respectively. We will use the all-floppy and all-rigid decompositions as baselines to compare KINARI's decompositions on real proteins.

These baselines may seem quite rudimentary, but they are quite powerful in showing that the higher level of sophistication built into our system provides provably better results. For example, single domain proteins such as Dihydrofolate Reductase (DHFR, see case study in Section 3.4.3) may be mostly rigid, with a small flexible region at the active site. In the KINARI v1.0 decomposition of the open conformation (1RA1), 93% of the residues are contained in the largest rigid cluster. For such cases, the all-rigid baseline might perform better than other methods which err toward a more flexible model.

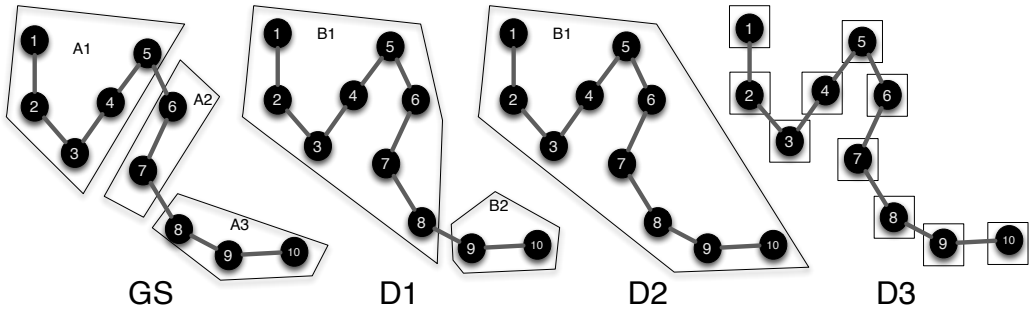


Figure 4.1: Three decompositions on the same example protein to demonstrate the B-cubed cluster decomposition score. GS represents the gold standard decomposition, and the rest are predicted decompositions. D2 and D3 are the all-rigid (100%-recall) and all-floppy (100%-precision) baselines. D1, D2, and D3 receive B-cubed scores, respectively, of 0.55, 0.46, and 0.65. See Table 4.1 for example calculations.

Example of calculated B-cubed scores. Figure 4.1 depicts the predicted decompositions for an abstract molecule, compared with a gold standard decomposition (GS). The decompositions are D1: produced by some predictive method, D2: all-rigid, and

D3: all-floppy. The B-cubed scores for D1, D2, and D3, compared with GS, are respectively 0.65, 0.55, and 0.46 (See Table 4.1 for calculation). This shows that the decomposition of D1, which visually seems to better match the cluster distribution of GS, achieves a higher score than either the all-rigid or all-floppy decomposition.

	D1		D2		D3	
Residue	*Re*	*Pr*	*Re*	*Pr*	*Re*	*Pr*
1	$\frac{5}{5}$	$\frac{5}{8}$	$\frac{5}{5}$	$\frac{5}{10}$	$\frac{1}{5}$	$\frac{1}{1}$
2	$\frac{5}{5}$	$\frac{5}{8}$	$\frac{5}{5}$	$\frac{5}{10}$	$\frac{1}{5}$	$\frac{1}{1}$
3	$\frac{5}{8}$	$\frac{5}{5}$	$\frac{5}{10}$	$\frac{1}{5}$	$\frac{1}{1}$	$\frac{5}{5}$
4	$\frac{5}{5}$	$\frac{5}{8}$	$\frac{5}{5}$	$\frac{5}{10}$	$\frac{1}{5}$	$\frac{1}{1}$
5	$\frac{5}{5}$	$\frac{5}{8}$	$\frac{5}{5}$	$\frac{5}{10}$	$\frac{1}{5}$	$\frac{1}{1}$
6	$\frac{2}{2}$	$\frac{2}{8}$	$\frac{2}{2}$	$\frac{2}{10}$	$\frac{1}{2}$	$\frac{1}{1}$
7	$\frac{2}{2}$	$\frac{2}{8}$	$\frac{2}{2}$	$\frac{2}{10}$	$\frac{1}{2}$	$\frac{1}{1}$
8	$\frac{1}{3}$	$\frac{1}{8}$	$\frac{3}{3}$	$\frac{3}{10}$	$\frac{1}{3}$	$\frac{1}{1}$
9	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{3}{3}$	$\frac{3}{10}$	$\frac{1}{3}$	$\frac{1}{1}$
10	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{3}{3}$	$\frac{3}{10}$	$\frac{1}{3}$	$\frac{1}{1}$
Avg	0.93	0.51	1.0	0.38	0.30	1.0
F1	0.65		0.55		0.46	
LRC-match	0.63		0.20		0.50	

Table 4.1: Calculation of B-cubed B-cubed precision, recall, and F1-scores for the small examples shown in Figure 4.1. The B-cubed score (shown as F1) for decomposition D1 is higher than the all-rigid (D2) and all-floppy (D3) decompositions. In the all-rigid baseline, all 10 residues are placed into the same cluster (D2), resulting in 100% recall but low precision. To contrast, in the all-floppy baseline, each residue is placed in a unique cluster resulting in 100% precision but low recall. The LRC-match score listed is the ratio of the sizes of the intersection and union of the largest rigid clusters from the two decompositions compared. Although not always the case, the LRC-match score correlates with the B-cubed scores for these three decompositions.

4.2.2 Benchmark data set

Gerstein Lab RigidFinder data set. To apply the cluster decomposition score evaluation method to real proteins, gold standard decompositions, to which compare the results, are needed. To serve this purpose, we have chosen a dataset from the

Gerstein Lab which is accompanied by decompositions that were validated against evidence from the biochemistry literature.

The Gerstein Lab's data set, listed in Table 5.3, was originally used to validate the RigidFinder method, which determines rigid cluster decompositions using two conformations of the same protein [1]. Although both RigidFinder and KINARI produce decompositions, RigidFinder requires two unique conformations as input, while KINARI requires only one. The RigidFinder data set has good coverage over small (fewer than 200 residues), medium (between 200 and 500 residues), and large (501 residues or more) proteins, and associated decompositions are readily available from the RigidFinder server website. Due to limitations in the PDB format, we have excluded GroEL-GroES from our study.

MSU-FIRST data set. In order to compare the new modeling options with previous results, we include four proteins used in the validation of the MSU-FIRST software [47, 48]. The four proteins are the lysine-binding protein (closed, 1LST; open, 2LAO), HIV-1 protease (closed, 1HHP; open, 1HTG), dihydrofolate reductase (open, 1RA1; closed, 1RX1; occluded, 1RX1), and adenylate kinase (open, 1DVR; closed, 1AKY). Although different PDB files were used, there is an overlap between these proteins and those in the RigidFinder data set because these are standard, well-studied proteins for which multiple conformations are known. We used RigidFinder to determine our 'gold standard' rigid cluster decompositions, choosing the decomposition according to the convention established in the paper [1], choosing the first sensitivity cutoff at a local maximum.

Converting to residue-level clusters. KINARI employs an all-atom model to determine rigid clusters of atoms, and not residues. Because the decompositions for the Gerstein Lab's benchmark data are at the residue level, the KINARI output must be transformed to a residue decomposition. In order to do this, we examine the body-bar-hinge model output by KINARI. For each atom-level rigid cluster determined by

KINARI, we first create an empty residue-level cluster and then collect the residues whose CA atoms belong to the cluster. Note that because rigid clusters can overlap, the CA atoms do not necessarily belong uniquely to that cluster. For each such CA atom, we examine the C-CA and CA-N bonds. If neither corresponds to a hinge in the body-bar-hinge model, meaning that the rotation is inhibited by the network of chemical constraints, we add the CA atom's residue to the residue cluster. Finally, the residue-level cluster is added to the decomposition to be compared with the benchmark.

4.2.3 Benchmarking toolkit

We have developed scripts for benchmarking systems which produce rigid cluster decompositions of proteins. This is a general framework that could be applied to other systems, not just those that rely on pebble game rigidity analysis.

- scoreRigidityResults.py : takes as input a gold standard and predicted decompositions and outputs the B-cubed precision, recall, and F1-score
- getNaiveBCubedScores.py : takes as input a gold standard decomposition and outputs the B-cubed precision, recall, and F1-score for each of the all-floppy and all-rigid baseline decompositions.

The ingredients for using the benchmarking toolkit are (1) a data set of PDBs with some associated gold standard cluster decompositions and (2) predicted decompositions on the same data set of proteins. As input, the scripts support the file format produced by RigidFinder for defining the decompositions into sets of residues.

The scripts, as well as the data sets, are available for download from the KINARI website (`http://kinari.cs.umass.edu/Downloads/benchmarking/`).

4.3 Results

We present the results of our evaluation of KINARI v1.0 on the benchmark data set. We used the RigidFinder server to generate a residue-level cluster decomposition for each of the proteins, using the two conformations for each protein as input. We then computed the B-cubed scores for KINARI v1.0 decompositions in order to score how well the decompositions matched those of RigidFinder. For comparison, we also computed the B-cubed scores for the all-floppy all all-rigid baselines. Table 4.2 lists the results of our evaluation using the B-cubed evaluation method for the two baselines and KINARI v1.0.

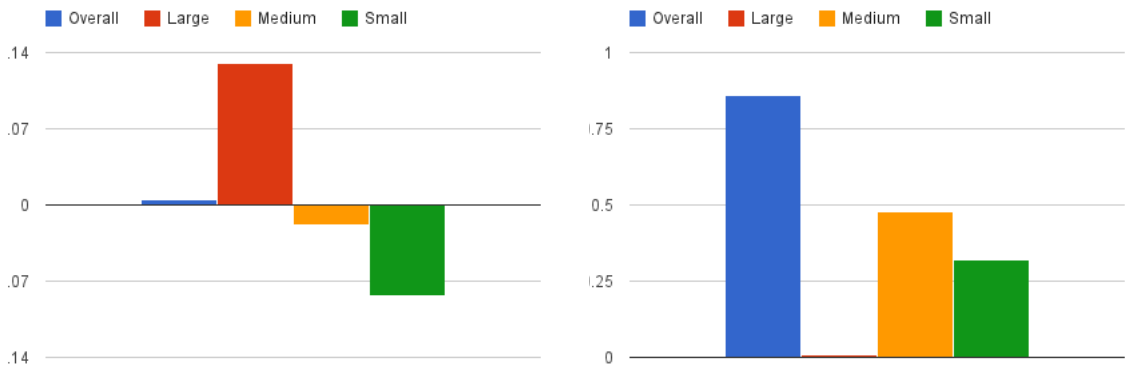
First, we discuss the results on the MSU-FIRST data set. For 3 of the 4 proteins, KINARI v1.0's score matched or performed better than the all-rigid baseline, for either the open or closed conformations. The RigidFinder decomposition for Dihydrofolate Reductase is quite rigid, with over 90% of residues lying in the largest rigid cluster. Although KINARI v1.0 detected the flexible loops for 1RA1 (see previous chapter, Section 3.4.3), the RigidFinder decomposition detected only one of them. A different tuning value for the RigidFinder method, with a more flexible decomposition, may result in improved B-cubed scores for KINARI v1.0.

Next we discuss the results on the 32 PDBs in the Gerstein lab data set. For 11 of the 17 proteins, the B-cubed score for at least one of the conformations was higher than that of the completely rigid baseline. For some of the proteins, there was a large discrepancy between the B-cubed scores. For example, the closed (1kc7) and open (2r82) conformations of pyruvate phosphate dikinase received scores of 0.45 and 0.66, respectively.

We performed a paired t-test on the results from the RigidFinder data set in order to evaluate whether improvement was significant over the all-rigid baseline. The means of differences and p-values are shown in Figure 4.2. This was indeed the case for the large proteins in the set (p-value, 0.0077), but overall, the improvement

over the baseline was not statistically significant. For the PDBs of medium and small proteins, the mean of the differences in B-cubed was negative, showing that a completely rigid decomposition was a better method prediction of the compared with the gold standard decomposition. For example, the KINARI v1.0 decomposition on Antigen 85C pdb 1dqz received a B-cubed score of 0.89, the highest among the Gerstein Lab data set. But method 2 received an even higher score, 0.92.

In summary, KINARI v1.0 can produce significant results for large proteins, but for medium and small proteins in the data set, the results are not significant. In the next chapter, we explore different parameterizations of the rigidity analysis and how these may improve accuracy.



(a) mean of differences

(b) p-value

Figure 4.2: Comparison of KINARI v1.0 B-cubed scores against all-rigid baseline. The B-cubed scores for the KINARI v1.0 rigid cluster decompositions and the all-rigid decompositions were computed for the RigidFinder data set, as shown in Table 4.2. The mean of differences measures the change in B-cubed score between the two methods. The p-value indicates whether the improvement is significant. We use the convention that a p-value of 0.05 or less is significant.

			1	2	3
			all floppy baseline	all rigid baseline	KINARI v1.0
Protein	Size (#Res)	PDB	B-cubed score	B-cubed score	B-cubed score
MSU-FIRST Data Set					
HIV-1 protease	198	1HHP	0.03	0.72	0.72
		1HTG	0.03	0.72	0.71
Dihydrofolate Reductase	159	1RA1	0.03	0.94	0.92
		1RX1	0.03	0.94	0.85
		1RX6	0.03	0.94	0.82
Adenylate Kinase	220	1AKY	0.06	0.68	0.65
		1DVR	0.06	0.68	0.74
Lysine-binding protein	238	1LST	0.03	0.72	0.90
		2LAO	0.03	0.72	0.65
RigidFinder Data Set					
Pyruvate phosphate dikinase	872	1KC7	0.02	0.43	0.45
		2R82	0.02	0.43	0.66
T7 RNA polymerase	843	1QLN	0.25	0.53	0.62
		1MSW	0.25	0.53	0.57
RNA polymerase II	3519	1I50	0.11	0.60	0.70
		2NVQ	0.11	0.59	0.55
Nitrogenase	3074	1M1Y	0.01	0.62	0.87
		2AFI	0.01	0.62	0.67
Rhodopsin	627	1F88	0.20	0.26	0.58
		3CAP	0.22	0.26	0.48
Phosphotransferase	214	2ECK	0.07	0.57	0.41
		4AKE	0.07	0.57	0.41
Bacteriorhodopsin	170	1BRD	0.13	0.53	0.60
		2BRD	0.41	0.38	0.56
DNA polymerase beta	328	2FMQ	0.07	0.54	0.57
		9ICI	0.07	0.54	0.61
Alcohol dehydrogenase	374	6ADH	0.07	0.63	0.46
		8ADH	0.07	0.63	0.66
Malate dehydrogenase	333	1BMD	0.11	0.68	0.69
		4MDH	0.13	0.67	0.66
Antigen 85C	280	1DQY	0.07	0.91	0.88
		1DQZ	0.05	0.92	0.89
Aspartate aminotransferase	401	1AMA	0.02	0.72	0.68
		9AAT	0.02	0.72	0.66
S100A6	89	1K9K	0.13	0.34	0.35
		1K9P	0.13	0.34	0.65
Cro repressor	61	5CRO	0.09	0.88	0.45
		6CRO	0.06	0.90	0.47
HIV-1 protease	99	4HVP	0.04	0.75	0.52
		3HVP	0.04	0.75	0.49
Calmodulin	141	1CLL	0.19	0.56	0.55
		1CTR	0.15	0.57	0.48
Bungarotoxin	74	1IDG	0.41	0.31	0.46
		1IDI	0.41	0.31	0.46

Table 4.2: B-cubed scores for KINARI v1.0. The MSU-FIRST data set consists of 4 proteins used to evaluate the MSU-FIRST software [48]. The RigidFinder data set is categorized, from top to bottom, into large (greater than 500 residues), medium (between 200 and 500 residues) and small (fewer than 200 residues) proteins.

4.4 Discussion

Performance of KINARI v1.0 on medium and small-sized proteins. The discrepancy in the performance of KINARI v1.0 on the larger and smaller proteins may be attributed to the noise in the set of hydrogen bonds and hydrophobic interactions. In Chapter 6, we will present our characterization of redundancy of noncovalent interactions, as identified by KINARI v1.0, in rigid clusters. Larger clusters have more redundant interactions which do not cause a loss of rigidity when they are removed. They are also less likely to contain interactions which are highly critical, such that when they are removed the cluster's size is reduced by 10% or more.

Gold standard data sets. A major challenge is in finding a very realistic gold standard data set. Within the RigidFinder data set which we used as our 'gold standard' decompositions, we found that there is some disagreement in the rigid and flexible domains as compared with those annotated by the authors of MSU-FIRST [48] (see Chapter 3, Section 3.4). RigidFinder is very effective at determining course-grained domain decompositions, but lacks the sensitivity to identify smaller flexible loops that may be functionally very important. This issue can be observed in decompositions of Adenylate Kinase (1AKY, 1DVR), which has 6 functionally important flexible loop regions as annotated in [48] (see also Table 3.4 and case studies in Chapter 2, Section 3.4). RigidFinder places the loops into their adjacent rigid clusters. With an improved annotated benchmarking data set, our evaluation should be repeated and the benchmark scores compared.

Atom-level decomposition. The evaluations were performed only on backbone flexibility, not taking advantage of the atom-level decompositions produced by KINARI. If such a benchmarking data set were available, it is within the power of our evaluation framework to compare sidechain flexibility as well.

4.5 Conclusion

The power of a benchmark lies in fast hypothesis testing. Beyond using case studies, in this chapter we demonstrated that KINARI v1.0 has significant predictive power over the all-rigid and all-floppy baseline for at least the large proteins in the data set. The performance on the larger sized proteins did not generalize for the smaller and medium-sized proteins. Especially for the smaller-sized proteins, the KINARI v1.0 predictions were overly flexible, and the all-rigid baseline performed better on most cases. These results lead us to question whether the parameterization of KINARI v1.0 is optimal for all proteins.

In the next chapter, we will propose new methods for modeling including hydrogen bonds and hydrophobic interactions in the modeling. We use our benchmarking methodology in order to validate the predictive power of the new methods.

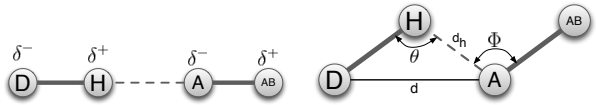
CHAPTER 5

ENERGY REFINED MODELING OF NONCOVALENT INTERACTIONS

In the previous chapter, we quantitatively measured KINARI v1.0's accuracy in predicting rigid cluster decompositions on a benchmark data set, and showed that with default parameter settings, the system outperformed the baselines for the large proteins in the data set. We also confirmed what has been observed in the literature, that there is no one-size-fits-all choice of curation and modeling options that will deliver good across-the-board performance. To support this claim, we showed that the all-rigid baseline proved to be a better predictor for the small and medium-sized proteins in the data set, perhaps due to inadequacies in the curation and modeling parameterization. In this chapter, we propose new approaches for incorporating hydrogen bonds and hydrophobic interactions into the modeling which can result in dramatic improvement in accuracy.

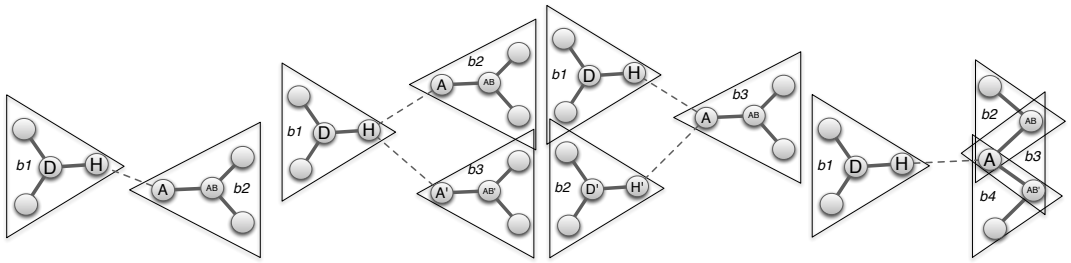
5.1 Introduction

Previous approaches. In previous work, hydrogen bonds were modeled as mechanically equivalent to covalent bonds, fixing bond length and bond angles at incident atoms [10, 21, 32, 48]. It has been observed early on that such a method may lead to inaccurate results, such as an almost complete rigidification of the protein model. Since it is known that not all hydrogen bonds have the same strength, an energy function was applied to prune the weakest bonds and exclude them from the model [48]. A universal hydrogen bond energy cutoff, which would produce biologically credible



(a) (b)

Figure 5.1: Definition of a hydrogen bond. (a) A hydrogen bond forms between a partially electronegative *acceptor* atom, A, and a hydrogen atom, H, that is covalently bonded to a partially electronegative *donor* atom, D. AB is the acceptor base. (b) hydrogen bonds are determined using geometric parameters.



(a) simple, non-furcated (b) furcated donor (c) furcated acceptor (d) multi-base acceptor

Figure 5.2: Hydrogen bond configurations. The triangles show the bodies determined by KINARI for the mechanical model.

results for any protein input, was never found. Wells *et al.* [94] point out the discrepancies of the hydrogen bond energy cutoffs in a number of previous studies in the literature.

In these systems, hydrophobic interactions were identified with heuristic approaches [10], and, unlike hydrogen bonds, they had no associated energies. It has been observed that the tuning of the hydrophobic interactions can be just as important as for hydrogen bonds. Gohlke *et al.* [32] comment, in their study of flexibility changes during Ras-Raf complex formation, "Finding the appropriate balance between these interactions [hydrogen bonds and hydrophobics] is thus crucial for an accurate representation of the flexibility characteristics of proteins".

Strength and geometries of hydrogen bonds and hydrophobic interactions. For covalent bonds, energy and geometry is characterized by the identities of the two

	1HTG	1VGC	1BBP
total	146	173	553
non-furcated D	144	169	521
bifurcated D	2	4	32
non-furcated A	124	136	416
bifurcated A	22	34	116
trifurcated A	0	3	21
single-base A	139	143	448
double-base A	7	11	33

Table 5.1: Frequency of hydrogen bonds which occur in special configurations (see Figure 5.2) in 3 example proteins.

electron-sharing atoms. The *bond length* and *directionality* (or *bond-angle*) tend to be fully determined, as explained by molecular orbital theory. For example, in ethane C_2H_6, each C atom is bonded to another C atom and 3 H atoms, forming two overlapping, rigid tetrahedra (Figure 3.3c). The bond angles and bond lengths remain relatively fixed. In mechanical modeling for rigidity analysis, covalent bonds are incorporated as bond-length and bond-angle fixing constraints. By contrast, hydrogen bonds display large variations in energies and geometries, even for those with the same donor and acceptor atoms [31] (page 1, paragraph 2). Strong hydrogen bonds behave essentially as a covalent bonds, but weaker hydrogen bonds behave more like electrostatic interactions which have much more variance in length and directionality [31]. Pairs of atoms which are packed closely together engage in hydrophobic (also called van der Waals) interactions. The strength of these interactions depends on the atom types and pairwise distances. Which hydrogen bonds and hydrophobic interactions to incorporate, and how to model them, is crucial to obtaining accurate results in rigidity analysis.

Contribution. We propose two new methods for incorporating noncovalent interactions for protein rigidity analysis. First, rather than simply removing weaker

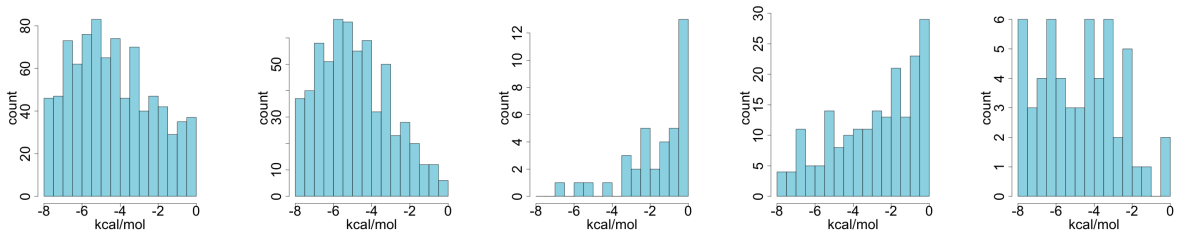
hydrogen bonds, we propose varying the way that they are modeled, based on their strength. We investigate modeling a weak hydrogen bond as a rigid bar which fixes the distance between the endpoints, but permits full rotational freedom. We reveal the limitations of the current mathematical theory for supporting this modeling, and propose heuristics to approximate the rigidity results. The second method we propose is in the inclusion of hydrophobic interactions. We calculate these interactions and assign to them an energy using the Lennard-Jones 6-12 potential. Then, as for hydrogen bonds, we use an energy cutoff to determine which interactions to include in the modeling. We investigated the use of a single, rigid bar to model these interactions. We have implemented these extensions in our KINARI software, and made it available for public use on the KINARI-Web server [21].

5.2 Background and Literature Review

Weaker hydrogen bonds play an important role in stabilizing a protein's fold and should not be neglected when modeling for rigidity analysis. Molecular mechanics forcefields support the calculation of hydrophobic, or van der Waals, interactions and their associated energies. In Sections 5.2.1 and 5.2.2, we give background on the biophysics of hydrogen bonds and hydrophobic interactions and how energies can be calculated. Because the new modeling methods, which we will propose in Section 5.3, may introduce degeneracies, we provide some relevant background material on this topic in Section 5.2.3.

5.2.1 Hydrogen bonds in proteins

A *hydrogen bond* forms between a partially electronegative *acceptor* (A) atom and a hydrogen atom that is covalently bonded to a partially electronegative *donor* (D) atom [52]. Schematically, we refer to the donor-hydrogen-acceptor triplet as D–H–A.



(a) All (b) Simple (c) furcated D (d) furcated A (e) multi-base A

Figure 5.3: The distributions of energies of hydrogen bonds varies based on its configuration. Hydrogen bonds from HIV-1 Protease (1HTG), Serine Protease (1VGC), and Bilin-binding Protein (1BBP)

Secondary structure elements in proteins, mainly α-helices and β-sheets, are held together by very regular hydrogen bonding patterns along the backbone [75]. Hydrogen bonds also form outside secondary structures, bracing together secondary structures and loops in the folded shape. Inter-molecular hydrogen bonds, such as those in the interface of two proteins in a complex, or between a protein and a ligand, play an important role in stabilizing the complex [15, 32, 73].

Strong, moderate, and weak hydrogen bonds. Hydrogen bond energies in proteins typically fall under 15 kcal/mol [31] (pg 31-32, Section 2.4.2). To compare, a typical covalent bond has an energy of 85 kcal/mol [75]. Figure 5.3a shows a histogram of the distribution of energies calculated in KINARI v1.0. The hydrogen bond energies are normally distributed with an approximate mean of -5 kcal/mol. The histogram shows that although fewer, the number of very weak hydrogen bonds, with energies near 0 kcal/mol is not negligible.

Weak hydrogen bonds are electrostatic in nature but increasingly behave like covalent bonds as their strength grows [31]. The boundary between 'weak' and 'strong' is blurred. One proposed assignment of a cutoff for weak bonds was < -4 kcal/mol [31]. For the modeling scheme proposed here, the boundary between weak and strong is left to the user to determine, as is the case in all previously implemented methods.

Hydrogen bond energy functions. Hydrogen bonds display large variations in energies and geometries, even for those with the same donor and acceptor atoms. This leads to difficulties in identifying bonds and their strengths, leading to what Gilli and Gilli [31] called the *hydrogen bond problem*. Despite the difficulty, efforts have been made to quantify the energy, or potential, of individual hydrogen bonds using orientation information alone. The two versions of FIRST and [10,48] and our own KINARI software use the Mayo energy function [68] for this purpose. This is a closed-form equation, parameterized on hydrogen bond angles and distances, as in Figure 5.1b. The energy function of Kortemme *et al.* is similarly parameterized by angles and distances, but rather than a closed-form equation, it sums the independent energetic contributions from a database of statistics from crystallography-determined protein structures [58].

Hydrogen bond configurations. Gilli and Gilli [31](pg. 24) present a review of a number of different configurations that have been studied in the hydrogen bond literature. Figure 5.2 shows several of them, which are relevant to our modeling. We use the nomenclature from [73] for describing furcated configurations.

In a *simple*, *non-furcated* configuration, the hydrogen atom, which is covalently bonded to a donor atom, forms a single hydrogen bond with an acceptor atom, which is also covalently bonded to only one atom, the acceptor antecedent, as in Figure 5.2a. A *furcated donor* configuration is formed when a single H engages in 2 or more hydrogen bonds. An acceptor which engages in multiple hydrogen bonds is in a *furcated acceptor* configuration. These three types of configurations have been identified and studied in the literature. One additional configuration that we identify here, which is not included in Gilli and Gilli's listing, is the *multiple-base acceptor* configuration where an acceptor that is covalently bonded to more than one atom engages in a hydrogen bond. Later, in the Methods section of this chapter (5.3), we describe why this special configuration is of concern in mechanical modeling.

The most stable configuration of a hydrogen bond is linear, with D–H–A (θ) forming a 180° angle, but hydrogen bonds are rarely found to be linear, and the most probable value is 165° [52] (pg. 20). Because of geometric constraints, hydrogen bonds in furcated configurations will tend to deviate even more from being linear.

Frequency of configurations in PDB data. Panigrahi and Desiraju [73] performed a survey on hydrogen bond configurations on a data set of structures of 251 protein-ligand complexes, using the HBAT software for determining hydrogen bonds. They found that overall 65% of acceptors and 34% of donors were in furcated configurations. Of the furcated acceptor configurations, 66% were bifurcated, 25% were trifurcated, and the remainder engaged in 4 to 6 hydrogen bonds. Of the furcated donor configurations, 39% were bifurcated, 27% were trifurcated, and the remainder were tetrafurcated, pentafurcated, or hexafurcated.

Configuration energies. We examine the 3 PDB files from [15] which probed the contribution of non-conventional hydrogen bonds to the rigidity of protein complexes: HIV-1 Protease (1HTG), Serine Protease (1VGC), and Bilin-binding Protein (1BBP). Table 5.1 shows the counts of types of hydrogen bonds in these three proteins. Because we use a different method to identify hydrogen bonds, the set of hydrogen bonds analyzed in Table 5.1 will have variations from those identified in [15]. In Figure 5.3, we collect together the set of hydrogen bonds from all three proteins, and show the distribution of energies for hydrogen bonds in different configurations. Hydrogen bonds in furcated configurations tend to be weaker than non-furcated because the angles tend to deviate further from 180°.

5.2.2 Hydrophobic interactions in proteins

A pair of neutral atoms are subject to two distinct forces between them: an attractive force at long ranges (van der Waals force), and a repulsive force at short ranges (Pauli repulsion force). The Lennard-Jones (or 6-12) potential, shown in

the equation below, is an approximation of the sum of these two forces involved in a hydrophobic interaction in proteins. The Lennard-Jones potential is standardly used in molecular mechanics force fields packages, such as the popular Amber-99 forcefield [8].

$$V = 4\epsilon(\frac{\sigma}{r}^{12} - \frac{\sigma}{r}^{6}) \tag{5.1}$$

The ϵ and σ values, which are the potential well depth and the distance at which the inter-atomic potential is zero, are experimentally determined and can be retrieved from tables distributed with the Amber-99 forcefield.

5.2.3 Non-generic models and rigidity theory

Tay's theorem applies to *almost all* geometric body-and-bar frameworks, but it fails on a statistically insignificant ("measure-zero") set of situations which are called *non-generic* due to the existence of certain algebraic dependencies between the geometric data. Identifying non-generic frameworks is in general a very difficult problem, but it is sometimes possible to state whether certain *combinatorially described* configurations are generic. A famous example is the *Molecular conjecture*, which states that molecular frameworks still obey Tay's theorem, generically, even when the set of hinges incident at an atom are concurrent. This conjecture, essential in establishing the validity of the combinatorial approaches for rigidity analysis of molecular structures, has been proven only recently [55], more than 25 years after it has been raised.

In this chapter, we discuss a number of situations, described in combinatorial (rather than geometric) terms and detected from the connectivity of the set of points and constraints, for which a *genericity theorem*, similar to the *molecular conjecture* would be needed. For practical purposes, we will have to work for now under the

assumption that the conjecture holds, as this is what allows for the extension of the pebble game algorithm to such cases; otherwise, our implemented method will have to be, for the time being, considered as an unproven heuristic. We point out, however, that such statements are sometimes notoriously difficult to prove; the first result of this kind, due to Maxwell (1864) and Laman (1970) took over 100 years to become a theorem. We will have to resort for now to empirical validation while waiting for the rigorous proofs.

To summarize: 3-dimensions only *generic body-bar-hinge frameworks* can be analyzed using Tay's theorem with theoretical guarantees of correctness. Moreover, in some situations the very definition of the associated graph fails to be well-defined. For example, when the endpoints of two bars coincide, there is no guarantee of them being in a generic position. We call this type of degeneracy a *bar-bar concurrency* (Figure 5.4b). If one endpoint of a bar lies on a hinge, we have a *bar-hinge concurrency* degeneracy (Figure 5.4c). How to place the edges in the associated graph for frameworks with this latter type of degeneracy is ambiguous. In the Methods section, we discuss where these degeneracies turn up in protein modeling, and propose a heuristic so that rigidity analysis can be performed.

5.3 Methods

In the next sections, we propose new methods for incorporating weak hydrogen bonds and hydrophobic interactions in the modeling. These methods will be evaluated in Section 5.4, using the data set and benchmarking methodology introduced in Chapter 4.

5.3.1 Modeling interactions with a bar

A bar placed between two bodies fixes distance but permits angles to vary. This bar concept is distinct from the 'tether' modeling introduced in ASU-FIRST, where

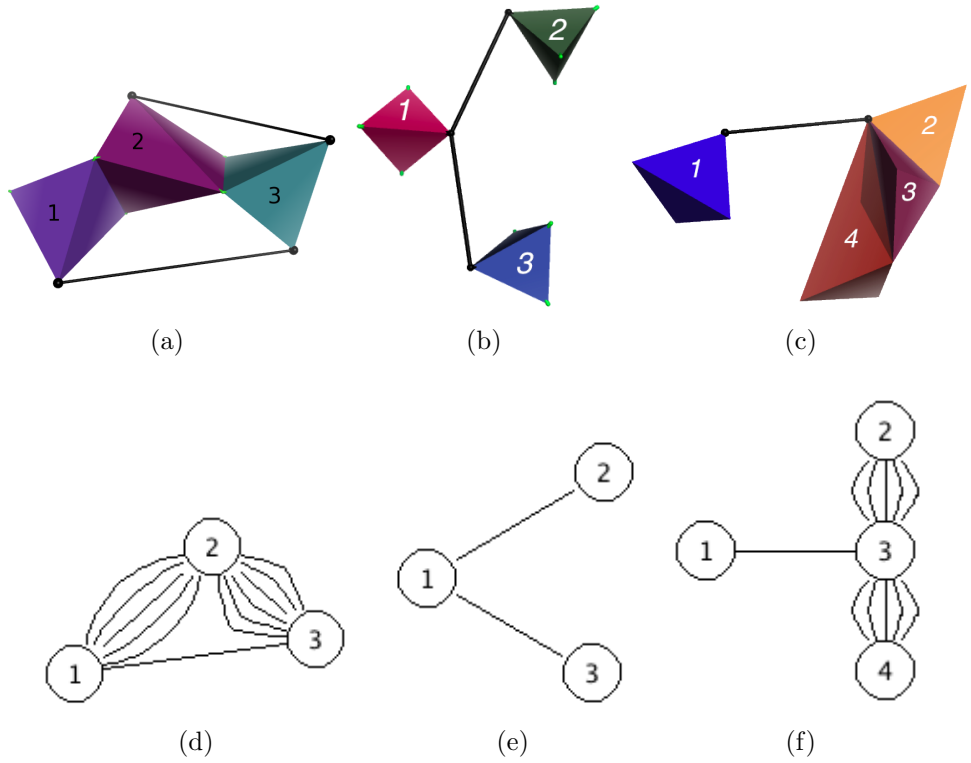


Figure 5.4: Examples of generic and non-generic body-bar-hinge frameworks and associated graphs. (a) shows a generic body-bar-hinge framework. The bar endpoints, and the continuous sets of points along the hinge axes, are all distinct. Its associated graph, shown in (d), is completely defined. The frameworks of (b,c) contain non-generic features described in this chapter: a bar-bar concurrency (b) and a bar-hinge concurrency (c). These two types of degeneracies may occur in mechanical models of proteins when modeling hydrogen bonds or hydrophobic interactions with a bar. Using our heuristic, we build associated graphs for the non-generic frameworks (e,f). Although for these two examples the pebble game will produce the correct result, there is no guarantee for non-generic cases.

bodies and multiple bars are placed in the model to approximate a pseudo-atom chain [10]. We are the first to investigate the use of this true type of bar for modeling. The motivation for using bars comes mainly from variable hydrogen bond modeling, as described next.

5.3.2 Modeling weak hydrogen bonds as bars

For our new modeling option for hydrogen bonds, we choose a cutoff energy value, but instead of discarding the weak bonds (as was done previously [48]), we model them with a weaker constraint than the one used for a covalent bond. The mechanical models produced may contain degeneracies, as discussed below. Furthermore, we survey where the problems occur for the different types of hydrogen bond configurations we defined in the previous section (Figure 5.2). For the different hydrogen bond topologies described earlier in the Background section, we describe how they are included in the mechanical modeling and the heuristics needed in order to build a graph for the pebble game algorithm. A case study is also provided to show how the different modeling can have subtle effects on the resulting rigid cluster decomposition.

Non-furcated configurations. Because H and A are both covalently bonded to only one other atom, during the body-building phase of modeling, they each are placed in one, and only one body. Placing the bar between the two bodies introduces no degeneracies because the endpoints are unique. No other bar is attached at the endpoint.

Furcated configurations. When the configuration is a furcated one, either at the hydrogen (Figure 5.2b) or acceptor (Figure 5.2c), then the mechanical model will contain two bars which share an endpoint. This bar-bar concurrency is combinatorially non-generic, but the multigraph associated to the resulting body-bar-hinge framework is well defined and the usual pebble game can be used in this situation.

Multiple-base acceptor configurations. An acceptor can be covalently bonded to multiple bases (Figure 5.2d). In the resulting mechanical model, A lies on a hinge and is in more than one body. The mechanical model will contain a bar-hinge concurrency, and the multigraph associated to the resulting body-bar-hinge framework is not uniquely defined.

Our heuristic. We propose a heuristic for building the associated graph in the combinatorially non-generic situations identified above. See the Background section for preliminaries. For each bar, if one of its end-points is a non-central atom, then it belongs to only one body and there is no ambiguity: we place an edge in the associated graph that is connected to the vertex corresponding to that body. If a bar's endpoint is attached to a central atom, then, in the multigraph associated to the mechanical structure, we place the edge on the vertex corresponding to the body of the central atom.

5.3.3 Calculating hydrophobic interaction energies and modeling as bars

This thesis is the first to evaluate the effect of varying the set of hydrophobic interactions. To provide a tuning parameter for inclusion, we are assigning an energy to each interaction based on its Lennard-Jones 6-12 potential. The ϵ and σ values are taken from the the Amber-99 forcefield [8]. Interactions with hydrogen atoms were excluded because these atoms take part in hydrogen bonding. Otherwise, all pairs of atoms, and not just those identified with the heuristic method for hydrophobics as introduced by ASU-FIRST, are considered as candidates for hydrophobic interactions. Figure 5.5 shows 51 hydrophobic interactions, with energies ranging between -0.15 and -0.2 kcal/mol calculated on an 18-residue α-helix. The previous version of our software, KINARI v1.0, would determine no hydrophobic interactions in the helix.

To model the hydrophobic interactions, we have chosen the single bar constraint described in the previous section on weak hydrogen bond modeling. This constraint models the atoms' propensity to stay a fixed distance from each other, while permitting angles to vary.

Using the new method for identifying hydrophobic interactions, a much greater number of hydrophobic interactions is identified. The number of hydrophobic interactions an atom might participate in varies by the energy cutoff used to exclude weaker

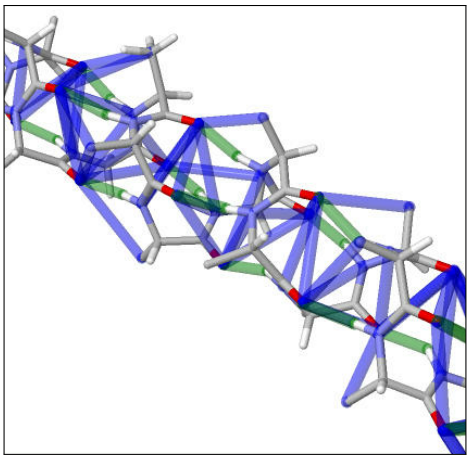


Figure 5.5: Hydrogen bonds (green) and hydrophobic interactions (blue) computed on a section of α-helix. In the 18 residue α-helix, 14 hydrogen bonds, with energies ranging between -2 and -7 kcal/mol, and 51 hydrophobic interactions, with energies ranging between -0.15 and -0.2 kcal/mol, were identified. With the heuristic version of calculating hydrophobic interactions in KINARI v1.0, no hydrophobic interactions would be identified within the α-helix.

interactions. See Figure 5.5 for an example of hydrogen bonds and hydrophobic interactions calculated by KINARI, where those with energies weaker than -0.15 kcal/mol are excluded.

5.4 Results and Discussion

We analyze now the results of applying the B-cubed score evaluation method to the benchmark data set. This builds on the evaluation results presented in the previous chapter. We have numbered the variants of the decomposition methods 1 through 7, as shown in Table 5.2. Refer to Chapter 4, Section 4.3 for the discussion of the evaluation on decomposition method 3, which is just KINARI v1.0 run with default options.

All scores and their associated cutoffs are listed in Table 5.3. The bar-plots in Figure 5.6 show the comparisons between each of the methods with method 2 (all-rigid decomposition) and method 3 (KINARI v1.0). The plots show the means of

Decomposition Method	Description
1	All-floppy decomposition
2	All-rigid decomposition
3	KINARI v1.0, default options
4	KINARI, vary hydrogen bond energy cutoff and exclude weak hydrogen bonds
5	KINARI, vary hydrogen bond energy cutoff and model weak hydrogen bonds as bars
6	KINARI, use default options for hydrogen bonds. compute hydrophobics and assign energy with LJ-potential. Exclude weak hphobes and model the rest as bars
7	same as Method 6, but vary the hydrogen bond energy cutoff and model the weaker hydrogen bonds as bars

Table 5.2: Evaluated rigid cluster decomposition methods.

differences and the p-values. We have also included two in-depth case studies on Pyruvate Phosphate Dikinase and Calmodulin, in order to demonstrate the sensitivity of the B-cubed evaluation method.

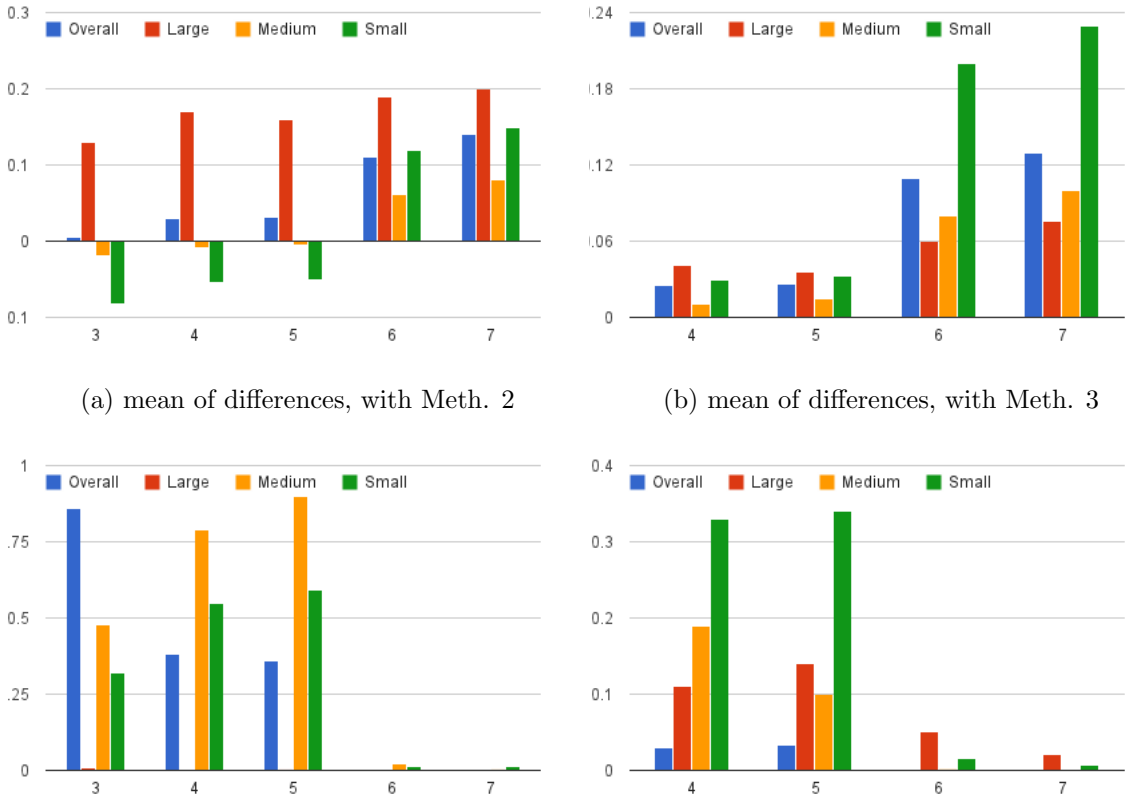
5.4.1 Cluster decomposition evaluation with decomposition methods 1 to 3, all-floppy and all-rigid baselines and KINARI v1.0

The decomposition methods 1 to 3 are (1) all-floppy baseline, (2) all-rigid baseline, and (3) KINARI v1.0 with default options. These two baseline decomposition methods are described in the previous chapter, Section 4.2.1. The first simply places each residue into its own rigid cluster, and the second places all residues into the same rigid cluster. The default options of KINARI v1.0 are shown in Tables 3.2 and 3.3 of Chapter 3.

We presented our evaluation of these three decomposition methods in the previous chapter. Although KINARI v1.0 proved to have non-trivial predictive power for determining RCDs on the larger proteins (> 500 residues) in the data set, the same version did not fare as well on the medium- and small-sized proteins. Overall, the

			1	2	3	4		5		6		7		
			all floppy baseline	all rigid baseline	KINARI v1.0	KINARI, remove weak H-bonds		KINARI, weak H-bonds as bars		KINARI, default H-bonds, vary Hydrophobics by energy		KINARI, vary both cutoff energies		
Protein	Size (#Res)	PDB	B-cubed score	B-cubed score	B-cubed score	B-cubed score	HB cutoff energy	B-cubed score	HB cutoff energy	B-cubed score	HP cutoff energy	B-cubed Score	HB cutoff energy	HP cutoff energy
MSU-FIRST Data Set														
HIV-1 protease	198	1HHP	0.03	0.72	0.72	0.73	-0.25	0.73	-0.25	0.74	-0.175	0.79	-2	-0.175
		1HTG	0.03	0.72	0.71	0.71	-0.75	0.71	-0.75	0.74	-0.15	0.77	-6	-0.15
Dihydrofolate Reductase	159	1RA1	0.03	0.94	0.92	0.92	0	0.92	-0.25	0.94	-0.125	0.94	-6	-0.125
		1RX1	0.03	0.94	0.85	0.86	0	0.86	-1.5	0.94	-0.125	0.94	-6	-0.125
		1RX6	0.03	0.94	0.82	0.82	0	0.82	0	0.94	-0.125	0.94	-6	-0.125
Adenylate Kinase	220	1AKY	0.06	0.68	0.65	0.65	-1	0.65	-2.25	0.68	-0.1	0.68	0	-0.1
		1DVR	0.06	0.68	0.74	0.74	0	0.74	0	0.78	-0.15	0.78	0	-0.1
Lysine-binding Protein	238	1LST	0.03	0.72	0.90	0.91	-1.75	0.91	-2	0.83	-0.2	0.83	-1	-0.2
		2LAO	0.03	0.72	0.65	0.65	-0.5	0.65	-0.75	0.94	-0.15	0.94	0	-0.15
RigidFinder Data Set														
Pyruvate Phosphate Dikinase	872	1KC7	0.02	0.43	0.45	0.66	-1.5	0.67	-2.75	0.6	-0.2	0.64	-6	-0.15
		2R82	0.02	0.43	0.66	0.66	0	0.66	0	0.66	-0.175	0.66	0	-0.175
T7 RNA Polymerase	843	1QLN	0.25	0.53	0.62	0.62	0	0.62	0	0.59	-0.15	0.63	-4	-0.15
		1MSW	0.25	0.53	0.57	0.57	0	0.57	0	0.65	-0.15	0.70	-4	-0.15
RNA Polymerase II	3519	1I50	0.11	0.60	0.70	0.7	0	0.70	0	0.66	-0.15	0.68	0	-0.175
		2NVQ	0.11	0.59	0.55	0.55	0	0.55	0	0.66	-0.15	0.67	-1	-0.15
Nitrogenase	3074	1M1Y	0.01	0.62	0.87	0.87	0	0.87	0	0.89	-0.15	0.89	-3	-0.15
		2AFI	0.01	0.62	0.67	0.72	-1	0.69	-2	0.88	-0.175	0.88	0	-0.175
Rhodopsin	627	1F88	0.20	0.26	0.58	0.6	-1.5	0.61	-3	0.57	-0.175	0.57	0	-0.175
		3CAP	0.22	0.26	0.48	0.61	-3	0.57	-5	0.59	-0.175	0.59	0	-0.175
Phosphotransferase	214	2ECK	0.07	0.57	0.41	0.41	0	0.41	0	0.57	-0.125	0.57	-7	-0.125
		4AKE	0.07	0.57	0.41	0.41	0	0.41	0	0.57	-0.125	0.60	-6.25	-0.15
Bacteriorhodopsin	170	1BRD	0.13	0.53	0.60	0.6	-2.75	0.64	-4	0.57	-0.125	0.67	-7	-0.125
		2BRD	0.41	0.38	0.56	0.66	-4.25	0.67	-5.75	0.63	-0.7	0.63	-0.75	-0.7
DNA Polymerase Beta	328	2FMQ	0.07	0.54	0.57	0.62	-0.5	0.62	-1	0.56	-0.175	0.63	-6.25	-0.15
		9ICI	0.07	0.54	0.61	0.61	0	0.61	0	0.71	-0.15	0.71	0	-0.15
Alcohol Dehydrogenase	374	6ADH	0.07	0.63	0.46	0.46	0	0.46	0	0.65	-0.15	0.67	-2	-0.15
		8ADH	0.07	0.63	0.66	0.66	0	0.67	-1.75	0.86	-0.175	0.86	0	-0.175
Malate Dehydrogenase	333	1BMD	0.11	0.68	0.69	0.69	-0.5	0.69	0	0.69	-0.15	0.72	-2.5	-0.15
		4MDH	0.13	0.67	0.66	0.66	-0.25	0.66	0	0.72	-0.15	0.74	-1.5	-0.15
Antigen 85C	280	1DQY	0.07	0.91	0.88	0.88	0	0.88	0	0.91	-0.15	0.92	-6.5	-0.15
		1DQZ	0.05	0.92	0.89	0.89	-1	0.89	-1	0.93	-0.15	0.93	-1.5	-0.15
Aspartate Aminotransferase	401	1AMA	0.02	0.72	0.68	0.68	0	0.68	0	0.74	-0.15	0.74	-1.5	-0.15
		9AAT	0.02	0.72	0.66	0.66	-1	0.66	-1	0.75	-0.15	0.75	-4.25	-0.15
S100A6	89	1K9K	0.13	0.34	0.35	0.63	-2.5	0.68	-2.5	0.62	-0.7	0.68	-2.25	-0.2
		1K9P	0.13	0.34	0.65	0.65	-1	0.65	-1	0.65	-0.7	0.74	-5.5	-0.175
Cro repressor	61	5CRO	0.09	0.88	0.45	0.45	0	0.45	0	0.95	-0.125	0.95	-7	-0.125
		6CRO	0.06	0.90	0.47	0.47	0	0.47	0	0.96	-0.125	0.96	-7	-0.125
HIV-1 protease	99	4HVP	0.04	0.75	0.52	0.52	0	0.52	0	0.75	-0.05	0.75	-7	-0.05
		3HVP	0.04	0.75	0.49	0.49	0	0.49	0	0.75	-0.075	0.75	-7	-0.075
Calmodulin	141	1CLL	0.19	0.56	0.55	0.56	-0.75	0.55	0	0.57	-0.175	0.62	-7	-0.15
		1CTR	0.15	0.57	0.48	0.48	0	0.48	0	0.81	-0.15	0.90	-7	-0.125
Bungarotoxin	74	1IDG	0.41	0.31	0.46	0.46	0	0.46	0	0.41	-0.15	0.41	-5.25	-0.15
		1IDI	0.41	0.31	0.46	0.46	-7	0.46	-7	0.41	-0.7	0.42	-7	-0.125

Table 5.3: B-cubed scores of each decomposition method on the benchmark data set. The MSU-FIRST data set consists of 4 proteins used to evaluate the MSU-FIRST software, and discussed in case studies in Section 3.4. The RigidFinder data set is categorized, from top to bottom, into large (greater than 500 residues), medium (between 200 and 500 residues) and small (fewer than 200 residues) proteins. Decomposition methods 1 through 7 are summarized in Table 5.2. Please see Figure 5.6 for a comparison of each method with the all-rigid baseline and KINARI v1.0.



(a) mean of differences, with Meth. 2

(b) mean of differences, with Meth. 3

(c) p-value, with Meth. 2

(d) p-value, with Meth. 3

Figure 5.6: Comparison of B-cubed scores on RigidFinder data set, as shown in Table 5.3. For each plot, the methods are numbered on the x-axis. In (a) and (c), methods 3-7 are compared with method 2 (all-rigid baseline). In (b) and (d), methods 4-7 are compared with method 3 (KINARI v1.0). The performance of each method is evaluated on the entire RigidFinder data set, as well as the large, medium, and small-sized protein subsets. The mean of differences measures the change in B-cubed score between the two methods. The p-value indicates whether the improvement is significant. We use the convention that a p-value of 0.05 or less is significant.

system was out-performed by the all-rigid baseline. In the next sections, we evaluate methods for tuning KINARI and measure how they may improve the system's performance.

5.4.2 Cluster decomposition evaluation with decomposition method 4, discarding weak hydrogen bonds.

For each of the PDBs, we compute the cluster decomposition score for the rigidity results produced at each hydrogen bond energy cutoff, excluding weaker hydrogen bonds. This is the conventional tuning parameter used in previous studies using rigidity analysis, as was first proposed for the MSU-FIRST software [48] (see also the discussion of cutoffs in [94]). For each PDB, the highest score was determined with its associated cutoff. If multiple cutoffs achieved the same score, the strongest (most negative) cutoff was the one used. The values are listed in Table 5.3.

For 11 of the PDBs (1HHP, 1RX1, 1LST, 1KC7, 2AFI, 1F88, 3CAP, 2BRD, 2FMQ, 1K9K, 1CLL), excluding weaker hydrogen bonds resulted in higher B-cubed scores than KINARI v1.0, in a few cases, quite substantially. This seemed to be the case when there was a large discrepancy in the KINARI v1.0 between two conformations of the same protein, as is typical for open and closed conformations. Removing hydrogen bonds from the more rigid conformation results in a decomposition that more closely matches those of RigidFinder and the other conformation in the pair. The case study of Pyruvate Phosphate Dikinase described in the next section will illustrate this phenomenon.

5.4.3 Cluster decomposition evaluation with decomposition method 5, modeling weak hydrogen bonds as bars.

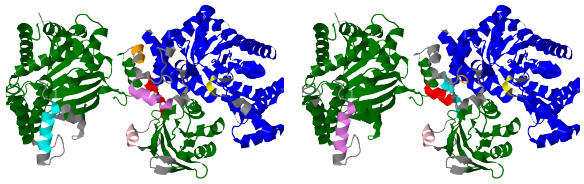
We reran our rigidity analysis experiments with KINARI, with the new proposed modeling method for weak hydrogen bonds described in the Methods section. For those PDBs for which using a cutoff did not lead to a higher score, the results were the same. For the 12 which benefited from the cutoff, 5 PDBs received higher scores, three did worse, and the rest remained unchanged.

The MSU-FIRST and Gerstein Lab's benchmark data sets of 21 proteins are insufficient for inferring general conclusions on whether the new modeling hydrogen bond method is significantly better than the default modeling method (removing weak hydrogen bonds). One of the tasks that should be undertaken in the future is to collect and validate a larger benchmarking data set.

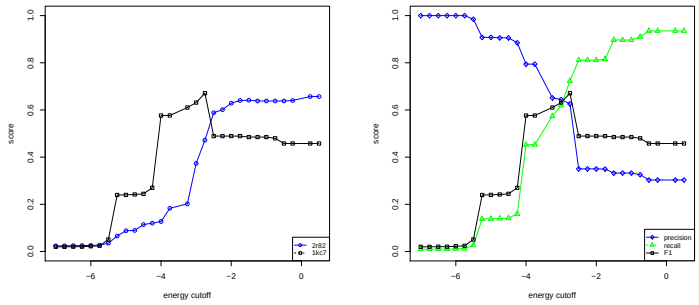
Hydrogen bond classification. Our hydrogen bond identification and classification method, using HBPLUS and the Mayo energy function, has its limitations. It would also be interesting to use different criteria to classify the bonds as weak and strong, for example, the duty cycle of Kurnikova *et al.* [61] or our classification of hydrogen bonds as critical and redundant [23].

Case study of Pyruvate Phosphate Dikinase. Pyruvate Phosphate Dikinase (PPDK) is a catalytic-enzyme which binds with ATP, pyruvate, and phosphate. The cluster decomposition produced by RigidFinder and KINARI v1.0 on the open (2R82) and closed (1KC7) conformations, are shown in Figure 5.7a, 5.7b, 5.7c. Visually, the 2R82 decomposition (Figure 5.7b) shows better agreement with that of RigidFinder's. A segmentation of the PEP/Pyruvate and His domains has been correctly identified. The ATP-grasp domain does not appear in its own cluster. The decomposition for the closed conformation (Figure 5.7c, 3D depiction of conformation not shown) placed most of the protein into the same rigid cluster. A small fragment, a single α-helix, of the ATP-grasp domain, has been determined to lie in a different rigid cluster. The two baseline decompositions, all-floppy and all-rigid, have scores of 0.02 and 0.43. Both the KINARI decompositions achieved better scores: 0.65 for 2R82 and 0.45 for 1KC7. The difference in scores between the two KINARI decompositions reflects the better accuracy of the decomposition for 2R82. The more rigid decomposition for 1KC7 is not surprising, given that it had 10% more hydrogen bonds and 14% more hydrophobic interactions than 2R82.

(a) RigidFinder (b) 2R82, KINARI v1.0 (c) 1KC7, KINARI v1.0



(d) 1KC7, method 4 (e) 1KC7, method 5



(f) B-cubed scores (g) 1KC7, precision, recall

Figure 5.7: RigidFinder and KINARI decompositions of Pyruvate Phosphate Dikinase. (a) RigidFinder decomposition. (b) The KINARI decomposition for 2R82 was optimal at cutoff energy 0, meaning all hydrogen bonds were included. For 1KC7, the maximum score, 0.66, was achieved at -1.5 kcal/mol when excluding weak hydrogen bonds from the modeling (d). By using a bar to model weak hydrogen bonds, a slightly better score (0.67, cutoff -2.75 kcal/mol) was achieved (e). The B-cubed score plots for the two conformations (f) when using the new modeling option. As the cutoff is varied, the precision and recall are monotonically increasing and decreasing. An optimal B-cubed score is achieved when the F1-values combining the precision and recall is optimized.

By excluding weaker hydrogen bonds from 1KC7, as in method 4, a decomposition which more closely matches the RigidFinder decomposition is achieved. Figure 5.7f shows the F1-scores for the decompositions produced by method 4 for each hydrogen

bond energy cutoff. For 2R82, the match is optimal at a cutoff of 0 kcal/mol, where all hydrogen bonds are retained. For 1KC7, excluding hydrogen bonds weaker than -1.5 kcal/mol results in the optimal score of 0.66. The corresponding decomposition is shown in Figure 5.7d, which places the important functional domains into separate rigid clusters. Applying method 5 (bar modeling) to 1KC7 results in a higher B-cubed score of 0.67 at the optimal cutoff of -2.75 kcal/mol, see Figure 5.7e.

This example on PPKD shows that by calculating the B-cubed score, the optimal cutoff can be determined automatically. Although there is no universal parameterization for rigidity analysis, future studies should explore under what conditions, such as conformational state or active temperature, the same cutoff best applies.

Prevalence of degeneracies in mechanical models. Because they are usually less linear, hydrogen bonds in furcated configurations tend to have weaker energies. They are more likely to be left-out from the mechanical model if an energy cutoff is used. Furcated bonds are bundled together (by definition), so removing them can have a drastic impact on the rigidity of a local area. By modeling them as a bar, we can more realistically capture them in the model as weaker then covalent bonds. Although individually the bars make a smaller contribution, when taken together, they have a significant effect on the rigidity.

Furcated configurations do introduce bar-bar concurrency degeneracies into the model, and depending on the boundary chosen between weak and strong, bar-bar degeneracies may be in abundance in the mechanical model. There is a sterically-imposed bound on the number of hydrogen bonds in a furcated configuration. Although in their study Panigrahi and Desiraju found examples of up to hexafurcated configurations, these were rare, and most configurations were bifurcated or trifurcated [73].

All bar-hinge concurrencies in the mechanical model are introduced when modeling hydrogen bonds in multi-base acceptor configurations. These hydrogen bonds tend

to be less common (about 6% of the hydrogen bonds in Table 5.1) and when they do occur, they are stronger (Figure 5.3e).

5.4.4 Cluster decomposition evaluation with decomposition method 6, when using hydrophobic interaction energy cutoff.

We repeated the evaluation, but this time, we used our new methods for hydrophobic interaction identification and modeling, as described in the Methods section. All identified hydrogen bonds were included and modeled with the default modeling option, but the hydrophobic interaction energy cutoff was varied.

For the RigidFinder data set, the improvement in the B-cubed scores over the baselines, methods 2 and 3, was significant. Compared with method 3, the mean of differences over the set of PDBs was 0.11 overall, and the p-value in the paired t-test was 0.00071 (Figure 5.6). The improvement was near-significant for the large proteins (p-value 0.051), and significant for the medium and small-sized proteins (p-values 0.0015 and 0.015). For the majority of proteins, the change in the hydrophobic modeling improved B-cubed scores. There was no consensus in the best energy cutoff value, but the median was -0.15 kcal/mol.

5.4.5 Cluster decomposition evaluation with decomposition method 7, varying both hydrogen bond energy and hydrophobic interaction energy cutoff

We next varied the energy cutoffs for both hydrogen bonds and hydrophobic interactions. Hydrophobics were included and modeled with the same scheme as in the method 6. For hydrogen bonds, we modeled bonds weaker than the cutoff with a 'bar' constraint (rather than excluding), as in method 5.

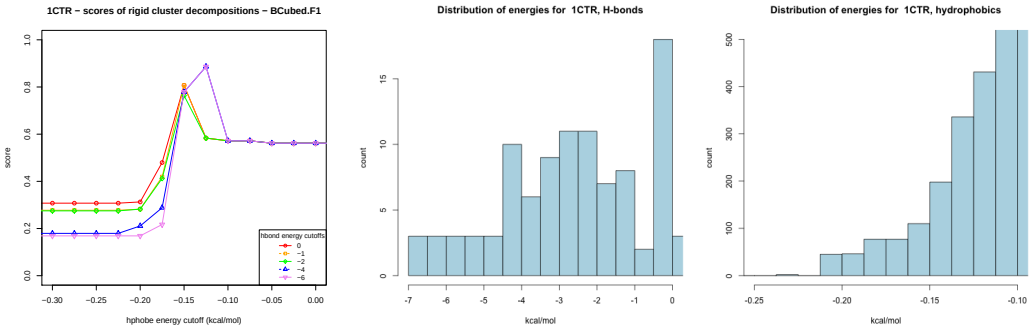
Compared with the highest scores attained on each PDB over all previous methods, method 7 achieved an improvement in over 70% of the 43 PDB files. The average change in score over method 6 was 0.02, and for a few cases, such as S100A6 and

Calmodulin, the increase in score was quite significant. As with the previous method, the median cutoff for hydrophobic interactions was again -0.15 kcal/mol. For the medium and large proteins in the data set, including some or all hydrogen bonds achieved the best score, confirming what has been stated in the literature [32] that the best results come from a balance between the two types of stabilizing interaction . For 4 out of 5 of the small proteins, including no hydrogen bonds (shown with a cutoff energy of -7 kcal/mol), but still excluding some hydrophobic interactions, produced the best decompositions.

We have introduced a method for identifying hydrophobic interactions and assigning energies based on the Lennard-Jones potential. This work is the first study to formally evaluate how the set of hydrophobics included can impact the rigidity results. Although it has been mentioned in papers from the Gohlke lab [29, 32] that the hydrophobic identification function was insufficient for achieving valid rigidity results for some classes of molecules (for example, RNA), there has been no thorough study in order to determine the best parameterization for hydrophobics. Ours is the first study to try to improve upon the inclusion of hydrophobic interactions.

Case study of Calmodulin. Figure 5.8d shows the results of varying both hydrogen bond energy and hydrophobic energy cutoff for Calmodulin (1CTR). By removing hydrogen bonds and adding hydrophobic interactions, an improved fit with the gold standard decompositions (shown in 5.8a) is achieved. Run with KINARI v1.0 (using its default options), the B-cubed score of 0.48 was worse than the score of the decomposition method 2 which is the all-rigid baseline (where all residues are placed into a single rigid cluster). By removing some or all hydrogen bonds and including some hydrophobic interactions, a score of 0.90 is achieved.

(a) RigidFinder (b) KINARI v1.0 (c) KINARI, new modeling

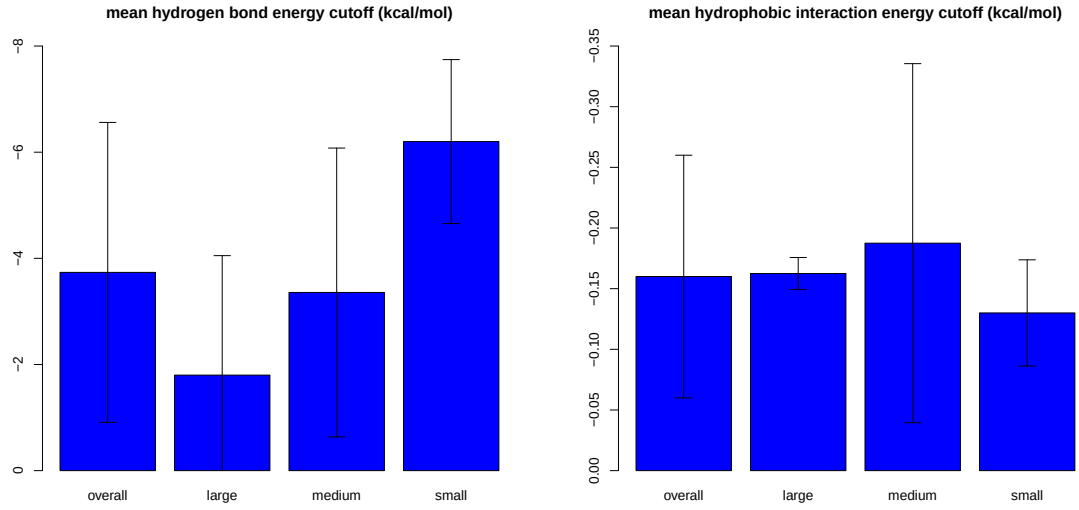


(d) B-cubed scores (e) hydrogen bond energies (f) hydrophobics energies

Figure 5.8: Accuracy of rigid cluster decompositions on Calmodulin (1CTR) improves with new modeling approaches. (a-c) Rigid cluster decompositions of 1CTR. (d) shows a plot of the B-cubed score as the energy cutoffs for hydrogen bonds and hydrophobic interactions are varied. (e-f) shows the distribution of hydrogen bond and hydrophobic interactions by energy.

5.4.6 Discussion

Toward automatic parameter settings in KINARI. In our evaluation of the decomposition methods, we have validated that accuracy can be improved by balancing hydrogen bonds and hydrophobic interactions in the modeling. How to optimally set rigidity analysis modeling options for an arbitrary protein remains an open question. We have shown in our evaluation that achieving a balance between hydrogen bonds and hydrophobic interactions can greatly increase the accuracy of RCDs. We were able to find the best tuning of parameter settings because the gold standard was available for comparison. When applying rigidity analysis as a predictive tool, no



(a) hydrogen bonds

(b) hydrophobic interactions

Figure 5.9: Mean optimal cutoff energy for hydrogen bonds and hydrophobics for decomposition method 7. The mean optimal cutoffs for the RigidFinder data set: overall, large, medium, and small are shown. Error bars show standard deviation.

such gold standard will be available. We examine our evaluation in order to make recommendations.

Figure 5.9 shows the mean optimal cutoff energies for hydrogen bonds and hydrophobic interactions computed in our evaluation of decomposition method 7 (see Section 5.4.5). (A more negative cutoff means fewer hydrogen bonds were classified as strong.) The mean hydrogen bond cutoff energies and protein sizes showed correlation. For the larger proteins, a weaker cutoff was advantageous, while for the smaller proteins, including more strong hydrogen bonds had less benefit. For hydrophobic interactions, there was no correlation between cutoff energy and size.

Future studies should validate whether using the B-cubed scoring and a training data set can assist in choosing the best modeling settings for protein families. For example, kinases are one of the larger protein families available from the Protein Data Bank, and often, multiple conformations are known. The benchmark data set (see

Section 4.2.2) includes multiple conformations of yeast and e. coli Adenylate Kinase. Future work should be undertaken to examine the variance of parameter settings across this protein family and others.

Evaluating the heuristic approach to handle degeneracies in the model. We have proposed a heuristic for placing the edges into the associated graph for non-generic bars in the mechanical model. To analyze, empirically and mathematically, when the heuristic works and when it fails is a problem for future investigations.

5.5 Conclusion

As has been iterated through the literature and demonstrated here, a one-size-fits-all parameterization for rigidity analysis does not deliver good across-the-board performance. Some tuning may be required to attain a rigid cluster decomposition for a protein that most closely agrees with data from experimental studies. We proposed two new methods: one for for inclusion and modeling of hydrogen bond and a second for the inclusion and modeling of hydrophobic interactions. We showed on a benchmarking data set that the new modeling in KINARI can produce rigid cluster decompositions, computed on single conformation, that better match 'gold standard' decompositions than previous methods. To do this, we applied a comparative decomposition scoring algorithm, first used in information retrieval, called the B-cubed score, and described in the previous chapter in Section 4.2.1.

CHAPTER 6

EVALUATING ROBUSTNESS OF RIGIDITY RESULTS

The focus of the previous chapter was on improving the accuracy in rigid cluster decompositions. Specifically, we measured how accuracy could be improved by tuning parameters using an energy cutoff. By measuring the precision and recall over a number of energy cutoffs, we gain insight into the sensitivity of the rigidity results as interactions are removed or modeled with a weaker constraint. For example, Figure 5.7g in the previous chapter shows a plot of the rate of change of the precision, recall, and B-cubed score during the tuning by hydrogen bond energy cutoff on PPKD (1KC7). This concept is similar to 'simulated unfolding' (or dilution), where the loss of rigidity is monitored while interactions are broken by order of energy. But this approach assumes that weaker interactions always break before their stronger counterparts, where in fact, there is some randomization to this process.

In this chapter, we take a different approach, which does not rely on energy calculations, to studying cluster sensitivity. We measure the tolerance of a cluster's rigidity to the loss of any interaction, strong or weak. In this way, we aim to characterize the robustness built into protein rigidity by nature.

6.1 Introduction

Atomic fluctations are essential for protein functions, such as ligand binding, because they permit the structure to adjust to the binding of another molecule [75]. The native state is stabilized by weak noncovalent interactions, namely hydrogen bonds (H-bonds) and hydrophobic interactions, which due to fluctuations, break and form

frequently. When existing weak interactions are broken, the released atomic groups can make new interactions of comparable energy, potentially resulting in conformational rearrangement. Protein structures continuously fluctuate about the equilibrium conformation observed in X-ray crystallography and NMR experiments. Therefore, when developing methods that rely on PDB structural data to predict protein rigidity and flexibility, it is crucial to assess how fluctuations may affect the results.

A simplifying assumption of this method is that the set of interactions is static. Yet, as demonstrated in molecular dynamics simulations, noncovalent interactions break and form rapidly, typically over nanoseconds [61]. An open question concerns the sensitivity and robustness of the rigidity results. When using a rigidity analysis system, what is our confidence in the rigid cluster decomposition determined? *If any particular interaction within a cluster were to break, would the cluster remain rigid, "shatter" into many smaller clusters, or would the flexibility increase, but only negligibly?*

In this chapter, we present our investigation of the prevalence of redundant and rigidity-critical interactions:

- *Redundant interactions.* How much redundancy is built into the network of interactions which hold together rigid clusters? What is the tolerance of a cluster to the loss of any particular interaction?

- *Rigidity-critical interactions.* How prevalent are non-redundant (*critical*) interactions? These are the interactions, which when broken, cause a non-negligible change in flexiblity that may effect function. How much will a cluster's size decrease when a critical interaction breaks?

Contribution. We address these question by proposing a method to classify noncovalent interactions. Based on their individual contribution to the rigidity of the cluster, they are labeled as either *redundant* or *critical*. In addition, we describe a

method for scoring clusters using the classification. The *criticality value* of the interaction is the change in cluster size upon the interaction's removal. We characterize what is the typical occurrence of redundant and critical interactions with an evaluation on a benchmark data set of over 120 proteins. We show with case studies that interactions with criticality values $\geq$0.10 tend to be concentrated in the same local region and their removal causes functionally relevant changes in rigidity. We make these methods available from the KINARI-Web server (http://kinari.cs.umass.edu) [21].

6.2 Literature Review

We include a short literature review to place this work in context. For other details of related work, please refer to Chapter 2.

The output of MSU-FIRST included a *flexibility index*, associated with each bond, to "characterize the degree of flexibility" [48]. We describe the MSU-FIRST flexibility index in further detail in the Results and Discussion Section, where we compare it with our method for scoring rigid clusters. Gohlke [32] extended the MSU-FIRST flexibility index from a single protein to an MD trajectory, and used it to show changes in flexibility during protein docking. Other flexibility indices have been proposed based on B-values from PDB files and normal mode analysis [53, 59, 91]. A related method, made available in ASU-FIRST and the Flexweb server is *dilution analysis* [78, 90]. It can be interpreted as a *simulated unfolding* because H-bonds are broken one-by-one, by order of energy. The rigid clusters of the protein are computed at each step, with the most stable part, called the *folding core* remaining at the end. Dilution was used to show that proteins undergo a rapid phase transition from rigid to floppy [78], to computationally identify the protein folding core [40], and compare patterns of rigidity within homologues [33, 94].

Dilution studies an ensemble of models which are hypothesized to reveal the unfolding path. Other efforts have been made to study an ensemble of around the native

state. Calculated from MD snaphots, the *duty cycle*, is the percentage of time a particular interaction is present. It has been used as a criterion for which interactions to include in a rigidity analysis [61]. More recently, Gonzalez *et. al* proposed a heuristic method, called the virtual pebble game, for predicting ensemble-averaged rigidity for a protein with fluctuating noncovalent interactions [35].

Differences from prior work. Our work proposed here is distinct from these previous studies because we perform an exhaustive study and do not resort to sampling. We are interested in finding degenerate cases– those which may not be revealed with a sampling approach. Rather than studying unfolding as was done with dilution, the goal in our current work is to better understand the rigidity and flexibility properties of the native state.

6.3 Materials and Methods

In this section, we present the methodology for *redundancy analysis*. We describe our two methods applied to a rigid cluster for, first, classifying redundant and critical noncovalent interactions and, second, calculating a redundancy score. Later in this chapter, we will present a survey and case studies applying our redundancy analysis methods, thus we present the datasets used in the evaluation. We have made the methods available on the KINARI-Redundancy server, an extension of the KINARI-Web server [21].

To apply redundancy analysis, we perform curation on the PDB file, select modeling options, and then compute the rigid cluster decomposition using KINARI. Refer to Chapter 3 for details on these steps. Described next are our two methods which work on each rigid cluster. Also included in this section is a description of the data set that will be used in our evaluation, and an overview of the features of the KINARI Redundancy server.

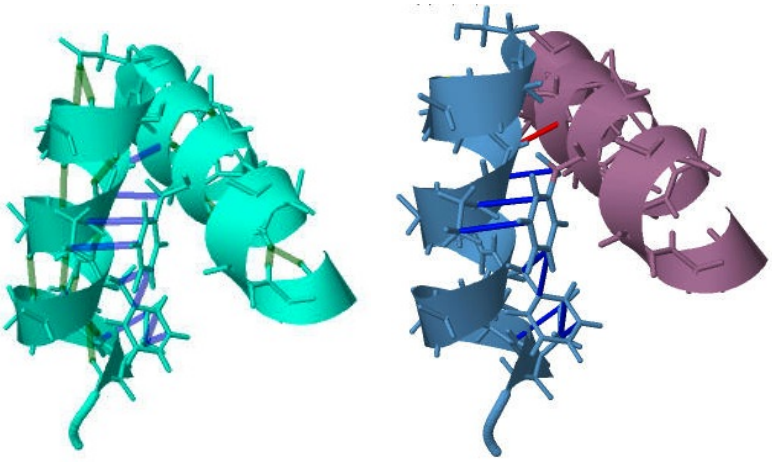
6.3.1 Identifying the critical and redundant interactions within a cluster

A rigid cluster is a maximal set of atoms and all bonds and interactions that hold them rigidly together. To identify the redundant interactions among the noncovalent interactions, we proceed as follows. One after another, we remove a noncovalent interaction, perform rigidity analysis, and verify if the cluster remains rigid, in which case, we classify it as *redundant*. Otherwise, the interaction is classified as *critical*. Note that once an interaction has been classified, it is placed back in the cluster. This approach is different from dilution, which removes interactions one after another, but does not replace them. Another difference is that hydrophobic interactions are not involved in dilution, only H-bonds, which have an associated energy.

To classify each interaction, we run the pebble game. In the worst case, this takes $O(n^2)$ time, thus the entire classification is in cubic time.

We propose a measure for each interaction, called its *criticality value*, based on how much the cluster size is impacted. We measure the size of the rigid cluster once an interaction has been removed and rigidity analysis re-run. The change in size of the cluster becomes the criticality value of the interaction. For example, the removal of an interaction with a criticality value of 0.10 will cause 10% of a cluster's atoms to break off into one or more separate clusters. We are interested in the cases in which high impact critical interactions occur.

To illustrate, see Figure 6.1, which shows the largest rigid cluster in Cytochrome-c (1HRC). The redundancy of this protein will be discussed in a case study in Section 6.4.2. The cluster is composed of two α-helices bound together by hydrophobic interactions (shown in blue). The hydrophobic interaction which is colored red is critical. When it is removed, each α-helix breaks off into its own rigid cluster. The *criticality value* of this interaction is 0.44.



(a) (b)

Figure 6.1: Hydrogen bonds and hydrophobic interactions in the largest rigid cluster of Cytochrome c (1HRC). (a) The cluster shown with H-bonds (green) and hydrophobic interactions (blue). (b) Removing the red hydrophobic interaction (with a criticality value of 0.44) from the cluster causes the two α-helices to split apart into separate rigid clusters (colors chosen at random).

6.3.2 Scoring of clusters by redundancy

Now we introduce a formula for the cluster redundancy score (Equation 6.1) which requires the classification of noncovalent interactions. $N(i)$ is the set of all noncovalent interactions in cluster i, and $R(i)$, the subset which are redundant. w_j is the weight assigned to an interaction j, determined by how it is modeled.

$$\Phi(i) = \frac{\sum_{j \in R(i)} w_j}{\sum_{k \in N(i)} w_k} \tag{6.1}$$

In this study, H-bonds and hydrophobic interactions have weights of 5 and 2, respectively, corresponding to the maximum degrees of freedom that may be removed by the interaction in the mechanical model. If all of the noncovalent interactions within the cluster were redundant, the redundancy score is 1. If instead all interactions were critical, the redundancy score is 0.

6.3.3 Data sets

We employ a few different data sets in this current study.

Multiple conformations. We have collected the PDB files of proteins published in the validation of the MSU-FIRST software [48]. The proteins are HIV-1 Protease (1HHP, 1HTG), Dihydrofolate Reductase (1RA1, 1RX1, 1RX6), Adenylate Kinase (1AKY, 1DVR), and Lysine-Arginine-Ornithine binding (LAO-binding) protein (1LST, 2LAO). We curated the protein data using the KINARI-Web curation tool. Ligands were removed for all but Adenylate Kinase structures. Hydrogen atoms and bonds and interactions were calculated with default options, as described in [21]. Since 1HHP is a homo-dimer, but only chain A is included in the PDB file, we applied a symmetry operation to compute the dimer [12]. Building the biological unit is available as an option from the KINARI-Web curation tool. We also employ a data set from the Gerstein Lab of 12 proteins used by the Gerstein Lab to validate the RigidFinder server [1]. We have excluded 5 of the proteins in the data set that contained more than 500 residues.

Proteins with known foldons. We include a case study of Cytochrome-c, a protein for which the foldons, intermediate structures which form during the folding process, are known [67].

Pdomain benchmark data set. For our survey to characterize the presence of redundant and critical interactions over a range of different proteins, we use the Pdomain Balanced Domain Benchmark 3 data set [42]. The original purpose for the data set was for benchmarking domain identification systems. We chose to use the data set because of the good coverage over the different topological groups as defined by CATH. We excluded 6 PDBs with clusters larger than 6000 atoms, so in total we included 121 PDB files in the analysis.

6.3.4 Redundancy Server

The redundancy analysis methods presented here serve as a tool for investigating the robustness of rigidity results, in particular, for users who wish to hand-edit their interaction set. We have deployed a server for redundancy analysis at the KINARI website. and KINARI-Mutagen tools [51]. Pre-processed examples and a video tutorial are provided to facilitate use. KINARI-Redundancy provides the following functionality:

- Curate PDB data and assign modeling options, as supported by the KINARI-Web server [21].
- Color clusters according to their redundancy score.
- Examine one cluster at a time in further detail. Color critical and redundant interactions.
- Filter the set of interactions displayed. A threshold can be selected to show only interactions with higher criticality values.

6.4 Results and Discussion

We applied our new methods on the data sets, described earlier in the previous section. We present the results of these in the next three subsections. Then, we compare the redundancy score with the flexibility index of MSU-FIRST [48] and present a discussion of future applications of our method.

6.4.1 Analysis of multiple conformations

We performed redundancy analysis on the multiple conformation data set described in Section 6.3.3. The rigidity analysis results of some of these proteins has been presented previously [48], and we seek out what more information the redundancy analysis can give us about these proteins.

Table 6.1 lists the results of our analysis. Most of the proteins had very few interactions with criticality values $\geq$0.10. We investigate these outliers in case studies on Adenylate Kinase, Dihydrofolate Reductase, DNA Polymerase β, and HIV-1 Protease.

		All		LRC	H-bonds in LRC					Hydrophobics in LRC				
Protein	PDB	Num. Residues	Num. Atoms	Num. Atoms	max crit. val.	total	with criticality value:			max crit. val.	total	with criticality value:		
							>0	>0.10	>0.25			>0	>0.10	>0.25
MSU-FIRST data set														
HIV-1 protease	1HHP	198	3126	1802	0.02	134	36	0	0	0.01	96	54	0	0
	1HTG	198	3126	1791	0.36	134	31	14	0	0.02	87	48	0	0
Dihydrofolate Reductase	1RA1	159	2484	1683	0.11	122	37	13	0	0.1	98	37	2	0
	1RX1	159	2484	1606	0.06	122	27	0	0	0.01	110	40	0	0
	1RX6	159	2484	1461	0.1	118	21	1	0	0.1	99	43	0	0
Adenylate Kinase	1AKY	220	3469	2032	0.12	176	51	5	0	0.01	89	29	0	0
	1DVR	220	3435	1787	0.17	144	25	2	0	0.01	124	36	0	0
LAO-binding protein	1LST	238	3554	1224	0.07	112	22	0	0	0.02	35	20	0	0
	2LAO	238	3608	1289	0.09	120	25	0	0	0.03	50	27	0	0
Gerstein Lab data set														
Bungarotoxin	1IDG	74	1084	59	0.44	3	3	3	3	0	7	0	0	0
	1IDI	74	1085	51	0	0	0	0	0	0.41	8	3	3	1
Calmodulin	1CLL	147	2185	1068	0.43	109	31	20	17	0.41	47	27	9	9
	1CTR	147	2139	456	0.26	41	12	2	2	0.06	19	5	0	0
Cro repressor	5CRO	61	958	324	0.02	24	5	0	0	0.03	22	6	0	0
	6CRO	61	948	306	0.43	23	10	6	3	0.09	29	7	0	0
HIV-1 protease	3HVP	198	1534	771	0.28	48	24	7	3	0.12	77	30	4	0
	4HVP	198	1534	638	0.36	48	26	19	9	0.04	38	18	0	0
S100A6	1K9K	90	1426	841	0.09	85	21	0	0	0.02	23	7	0	0
	1K9P	90	1435	338	0.49	39	9	2	2	0.49	10	7	6	6
Alcohol Dehydrogenase	6ADH	374	1757	1757	0.08	85	24	0	0	0.1	181	55	3	0
	8ADH	374	3819	3819	0.07	284	74	0	0	0.07	243	99	0	0
Antigen 85C	1DQY	282	3360	3360	0.02	269	43	0	0	0.02	306	51	0	0
	1DQZ	282	3118	3118	0.02	254	48	0	0	0.01	219	41	0	0
Aspartate Aminotransferase	1AMA	410	3576	3576	0.05	303	62	0	0	0.02	181	48	0	0
	9AAT	410	3383	3383	0.03	317	63	0	0	0.01	147	46	0	0
Bacterio-rhodopsin	1BRD	226	1818	1818	0.04	136	16	0	0	0.02	152	57	0	0
	2BRD	226	2148	2148	0.06	208	34	0	0	0.03	146	56	0	0
DNA Polymerase Beta	2FMQ	335	3106	3106	0.29	273	74	9	3	0.29	144	64	7	4
	9ICI	335	2336	2336	0.03	147	25	0	0	0.02	222	56	0	0
Malate Dehydrogenase	1BMD	332	3136	3136	0.08	292	52	0	0	0.01	146	53	0	0
	4MDH	333	2939	2939	0.04	220	51	0	0	0.04	192	69	0	0
Adenylate Kinase	2ECK	214	444	444	0.09	34	7	0	0	0.02	29	9	0	0
	4AKE	214	549	549	0.71	43	26	18	12	0.71	38	17	4	4

Table 6.1: Prevalence of critical and redundant interactions in the largest rigid clusters (LRCs) of the MSU-FIRST and Gerstein Lab data sets. For each PDB, the total number of residues and atoms are shown, as well as the size of the LRC. We classified each of the H-bonds and hydrophobics as either critical or redundant to its clusters rigidity. The number of interactions with criticality values $\geq$0.0, $\geq$0.0 and $\geq$0.0 are displayed. There were no interactions with criticality values $\geq$0.50 in the data set.

6.4.1.1 Adenylate Kinase

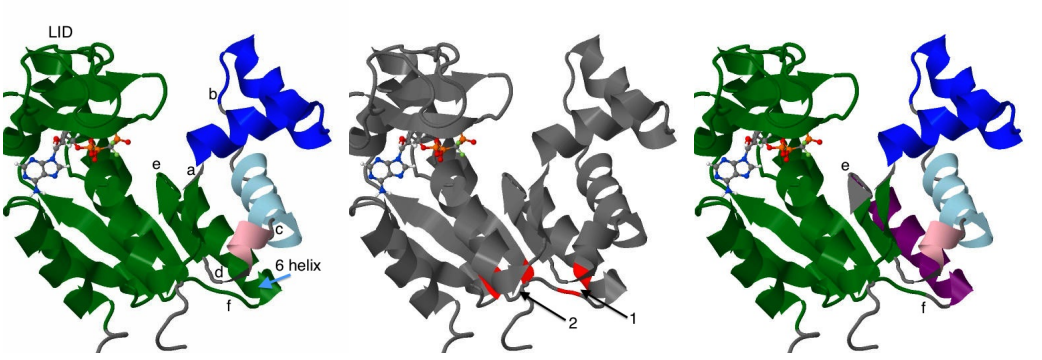
We first discuss the results on the yeast Adenylate Kinase (ADK), a monomer known to undergo domain-level hinge motion upon ligand binding. Figure 6.2 shows decompositions of ADK, depicted on the ATP-bound, open conformation (1DVR). The domain containing the binding site is labeled as the LID-domain. In early work of Jacobs *et. al* to validate the MSU-FIRST system, 6 flexible loops were detected in the open conformation (1DVR). 4 of the 6 were also detected in the closed conformation (1AKY). The 2 flexible loops not detected in 1AKY were those at the N- and C-terminals of the 6 α-helix (95 ILE to 108 GLN). The loops are labeled a-f in Figure 6.2.

Running redundancy analysis, two of the H-bonds (1 and 2) were found to decrease the size of the largest rigid cluster by 17% and 12% and are shown in Table 6.2. H-bond 1 lies near the N-terminal, at the end of the parallel β-sheet, and connects the β-sheet to the f-loop, between, 6 ARG O and 113 GLU H and H-bond 2 is between 105 LEU O and 110 THR H, anchors the end of the the 6 α-helix to the f-loop. Figure 6.2a shows the residues which engage in the two very critical interactions, highlighted in red. When either of these interactions is removed, the 6-α-helix breaks apart from the cluster and the e and f-loops become flexible. The rest of the LRC remains intact. These two H-bonds were assigned energies of -0.58 and -5.3 kcal/mol by the Mayo Lab energy function [68].

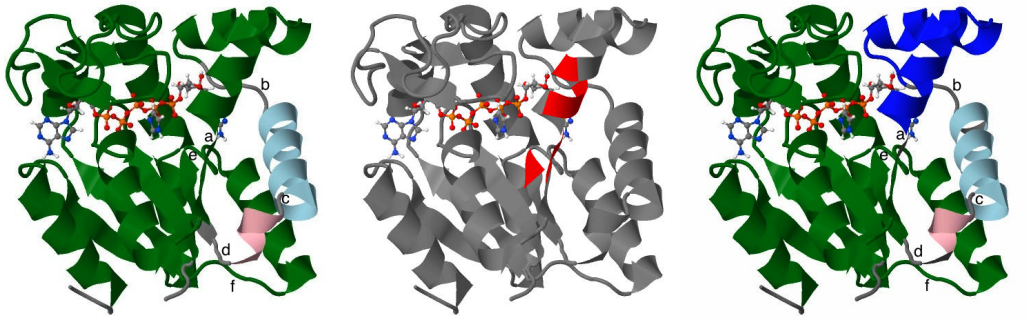
ID	Atom 1	Atom 2	Type	Energy	Crit. val.
1	6 ARG O	113 GLU H	HB	-5.3	0.17
2	105 LEU O	110 THR H	HB	-0.58	0.12

Table 6.2: Critical interactions in Adenylate Kinase (open, 1DVR). 2 H-bonds (HB) and 0 hydrophobic interactions (HP) with criticality values $\geq$0.10 were detected.

We investigated whether using a dilution would identify these critical interactions [78]. In dilution, or simulated unfolding, H-bonds are removed one-by-one and



(a) 1DVR, RCD, default (b) 1DVR, locations of critical H-bonds (c) 1DVR, RCD, without critical H-bond



(d) 1AKY, RCD, default (e) 1AKY, locations of critical H-bonds (f) 1AKY, RCD, without critical H-bond

Figure 6.2: Case study of Adenylate Kinase. (a-c), open conformation 1DVR. (a) The rigid cluster decomposition (RCD) determined by KINARI v1.0. The grey regions are flexible and each colored region is a rigid cluster. (b) H-bonds with criticality values of 0.17 (1) and 0.12 (2) were found in the largest rigid cluster (green). The residues which engage in these H-bonds are highlighted in red. (c) Upon removing H-bond 1, the e-loop and part of the f-loop becomes flexible. (d) Similarly, for H-bond 2, both e and f-loops gain flexibility, although the region of flexbility in the f-loop is closer to the β-sheet than the α-helix. (d-e), closed conformation 1AKY. (d) Rigid cluster decomposition with default options. (e) location of the 5 interactions which have the greatest impact on cluster size. (f) after removing a critical interaction.

cumulatively, by order of weakest to strongest, simulating H-bonds breaking during denaturation. Of the 146 H-bonds in the LRC, H-bond 2 was the 18th weakest with an energy of -0.58 kcal/mol. H-bond 1 with an energy of -5.3 kcal/mol ranked 109th, and would not have been revealed with a dilution analysis.

We next analyzed the closed conformation, 1AKY. With default options, KINARI predicts a larger LRC than for 1DVR, the open conformation. The MSU-FIRST software detected 4 flexible loop regions (b,c,d, and f) [48]. With default options, KINARI detects 3 of these loops (b,c,d). Our redundancy analysis detected 5 H-bonds with criticality values $\geq$0.12, listed in Table 6.3. When any of these interactions are removed, the resulting RCD contains a flexible region in the a-loop.

ID	Atom 1	Atom 2	Type	Energy	Crit. val.
1	34 ALA O	38 ALA H	HB	-3.96	0.12
2	37 ASP OD1	40 ASP HH21	HB	-4.78	0.12
3	40 ARG HH12	301 ARG O1E	HB	-2.22	0.12
4	40 ARG HH22	301 ARG O1E	HB	-4.68	0.12
5	33 LEU O	89 LEU H	HB	-4.89	0.12

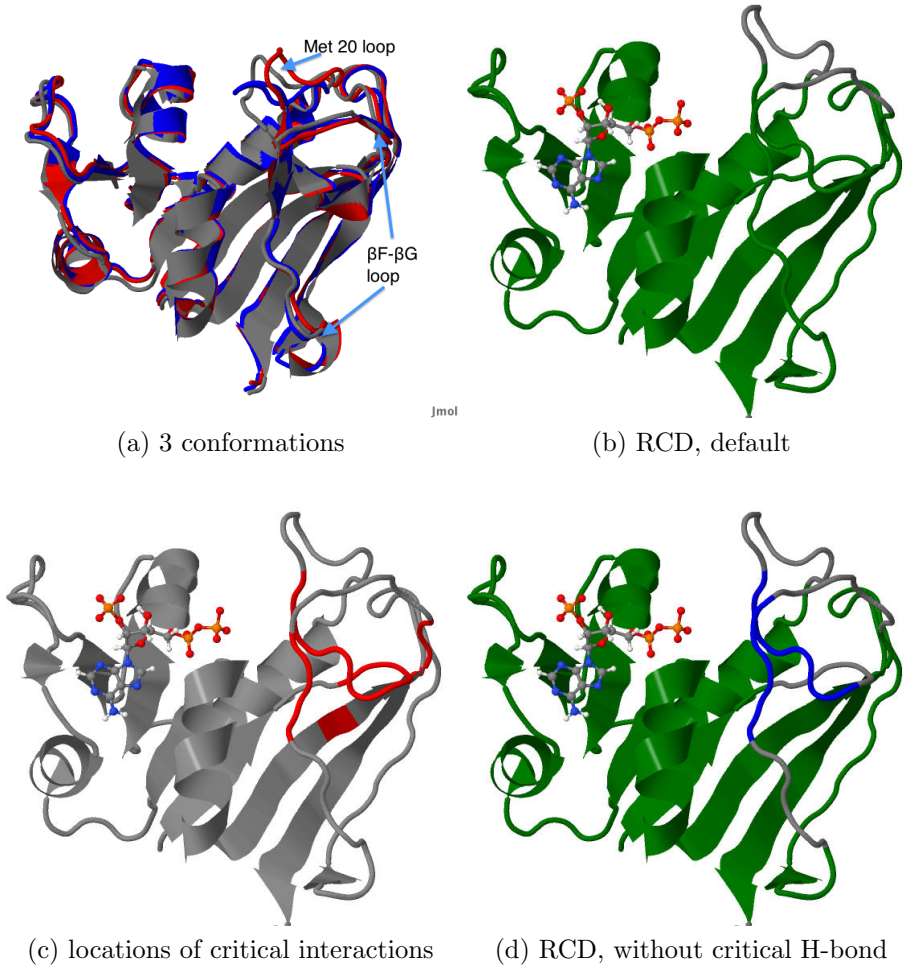
Table 6.3: Critical interactions in largest rigid cluster of Adenylate Kinase closed conformation (1AKY). 5 H-bonds (HB) and 0 hydrophobic interactions (HP) with criticality values $\geq$0.10 were detected.

To summarize, the LRCs of 1AKY and 1DVR both contained multiple interactions with criticality values $\geq$0.10, and these were concentrated together in the structures. In 1DVR, the removal of either of the two very critical interactions caused the same two loops to gain flexibility concurrently. In 1AKY, the removal of any of the 5 very critical interactions all caused the a-loop to become flexible. These loops were determined to be important to the flexibility and mobility for ADK to perform its function.

6.4.1.2 Dihydrofolate Reductase

1RA1, 1RX1, and 1RX6 are structures of respectively, the open, closed, and occluded conformations of e. coli Dihydrofolate Reductase (DHFR), a small enzyme which plays an essential role in DNA building. The flexibility of the Met20 loop (residues 9 - 24) and the βF- βG loop (residues 116-132) near the active site plays a

role in promoting the release of the product. Figure 6.3a shows an alignment of the three DHFR structures, demonstrating the high mobility in the Met20 loop.



(a) 3 conformations (b) RCD, default

(c) locations of critical interactions (d) RCD, without critical H-bond

Figure 6.3: Case study of Dihydrofolate Reductase. (a) A 3D alignment of three conformations (1RA1, 1RX1, 1RX6) shows the high mobility of the Met20 loop. (b) With default options, the rigidity results on 1RA1 show flexibility in the Met 20 loop, but the βF- βG loop is almost entirely rigid. (c) Residues which engage in very critical interactions in the largest rigid cluster of 1RA1 are almost all within the Met20 and βF- βG loops. (d) After removing a very critical interaction, a smaller cluster (blue) composed of part of the Met20 and βF- βG loops breaks off from the LRC, and a region in the βF- βG loop becomes flexible

The KINARI rigid cluster decompositions of the three structures show the protein to be mostly rigid, with most of the protein contained in the LRC. Each of the decompositions show flexible regions in the βF- βG loop. In 1RX1, the closed con-

formation, the Met20 loop is determined to be locked- it is contained in the largest rigid cluster. 1RA1 and 1RX6, the open and occluded conformations, regions of the Met20 loop are determined to be flexible, but there is some variation between the two RCDs. These results agree with earlier results of Jacobs *et al.* using the MSU-FIRST rigidity analysis software [48].

Table 6.1 includes the results of our redundancy analysis on the three conformations. The closed conformation contained no interactions with criticality values $\geq$0.10 and the occluded conformation contained only one. In the open conformation (1RA1), over 10% of the H-bonds (13 of 122), and 2 of the hydrophobic interactions had criticality values greater than 0.10. These very critical interactions are all concentrated around the active site, adjacent to the Met20 and βF-βG. Table 6.4 lists the set of very critical H-bonds and hydrophobic interactions and their locations. The majority (7 of the 13) of very critical H-bonds connect the Met20 and βF-βG loops. The remainder of H-bonds are in the local area of the two mobile loops. Removing any of these H-bonds increases the extends the flexible regions in the Met20 or βF-βG loop. None of these very critical interactions involve the ligand.

For example, when we remove the H-bond between atoms 8 LEU H and 113 LEU O, the flexible region of the Met20 loop and βF-βG loop increases substantially, as depicted in Figure 6.3d, even though this particular H-bond does not involve residues within those loops.

The LRCs of 1RX1 (closed) and 1RX6 (occluded) contain few to no H-bonds with high criticality values. These structures already had more flexibility in the βF-βG loop.

The prevalence of very critical interactions in the active site region for the open conformation shows that the rigidity of the βF- βG loop is 'hanging by a thread'. These H-bonds tended to be strong, and mostly backbone-backbone H-bonds, so

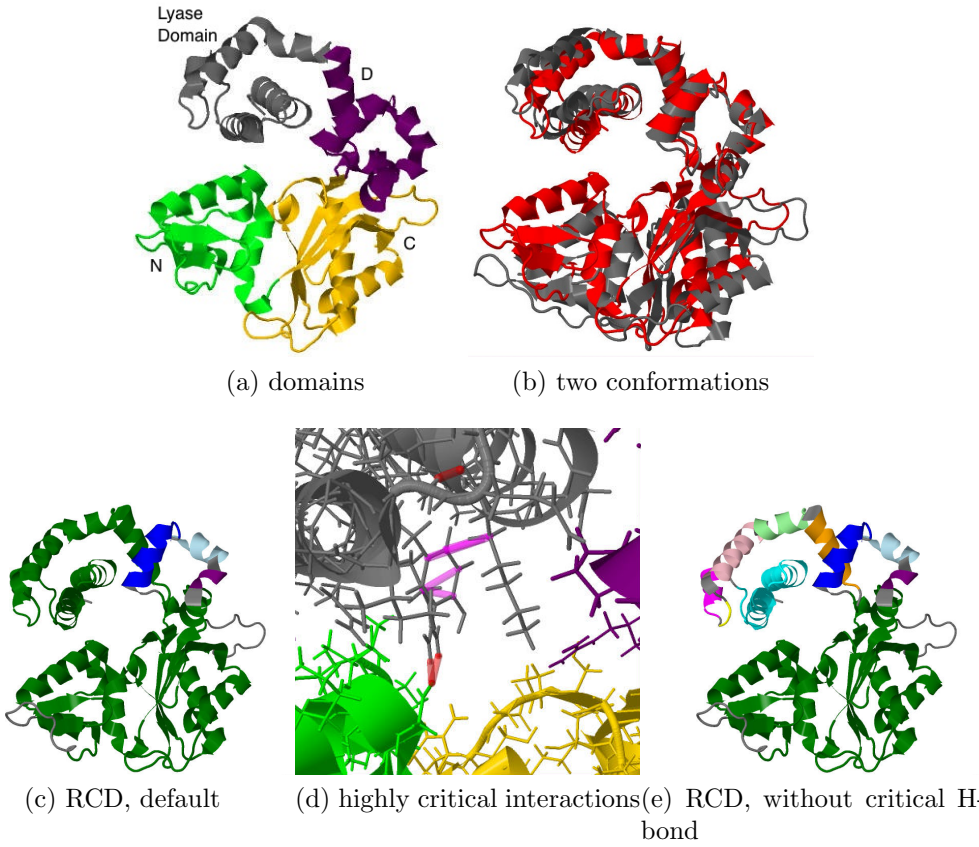
ID	Atom 1	Atom 2	Type	Ener.	C. V.	In Met20 or βF- βG loops?
1	8 LEU H	113 LEU O	HB	-4.84	0.10	Neither
2	116 ASP H	150 ASP O	HB	-7.07	0.10	Both in βF- βG loop
3	8 LEU O	115 LEU H	HB	-5.58	0.10	Neither
4	115 ILE O	117 ILE H	HB	-1.74	0.10	One in βF- βG loop
5	9 ALA H	13 ALA O	HB	-5.77	0.10	Both in Met20 loop
6	10 VAL O	13 VAL H	HB	-2.69	0.10	Both in Met20 loop
7	10 VAL H	117 VAL O	HB	-6.33	0.10	Met20 and βF- βG loop
8	12 ARG O	125 ARG H	HB	-5.4	0.10	Met20 and βF- βG loop
9	12 ARG HE	125 ARG O	HB	-6.79	0.10	Met20 and βF- βG loop
10	14 ILE H	123 ILE O	HB	-6.91	0.11	Met20 and βF- βG loop
11	15 GLY O	122 ASP H	HB	-1.81	0.11	Met20 and βF- βG loop
12	15 GLY O	123 GLY H	HB	-2.02	0.11	Met20 and βF- βG loop
13	15 GLY H	123 GLY O	HB	-2.38	0.11	Met20 and βF- βG loop
14	11 ASP C	12 ARG CG	HP	N/A	0.10	both in Met20 loop
15	123 THR C	124 HIS CG	HP	N/A	0.10	Met20 and βF- βG loop

Table 6.4: Critical interactions in Dihydrofolate Reductase (1RA1). 13 H-bonds (HB) and 12 hydrophobic interactions (HP) with criticality values $\geq$0.10 were detected. The type, energy, and criticality value for each interaction are shown. Also, the location of both atoms, with respect to the Met20 and βF- βG loops. Any of these interaction, when removed, lead to increased flexibility in the βF- βG loop, whether the two atoms were located within the loops. See also Figure 6.3.

independently, each is unlikely to break. But snipping any of these very critical interactions will cause the βF-βG loop to gain flexibility.

6.4.1.3 DNA Polymerase β

DNA polymerase β (POLB) is a 335 residue DNA and metal binding enzyme, responsible for base excision repair of DNA. It is active as a monomer and composed of an N-terminal 90-residue lyase domain connected to a C-terminal polymerase domain, composed of 3 subdomains [3]. The lyase domain and 3 subdomains are depicted in Figure 6.4a.



(a) domains (b) two conformations

(c) RCD, default (d) highly critical interactions (e) RCD, without critical H-bond

Figure 6.4: Case study of DNA Polymerase β (2FMQ). (a) 4 important functional domains. The lyase domain (grey) is the important catalytic site. The N domain interacts more strongly with the lyase domain in the closed conformation. (b) 3D structural alignment of 2FMQ (red) and 9ICI (grey). (c) The rigid cluster decomposition on 2FMQ shows one dominant rigid cluster (green) containing both the lyase and N domains. Grey regions are flexible. (d) If any of the H-bonds (red) and hydrophobics (pink) shown are removed, the lyase domain will break off from the dominant rigid cluster, as shown in (e). Note that only 2 of these interactions actually crossbrace between the lyase domain and the other domains in the protein, and the rest lie completely in the lyase domain. Therefore, a loss of an interaction within the lyase domain will cause the separation of the lyase domain from the rest of the cluster.

We examine the redundancy of one conformation of POLB, 2FMQ. The rigidity analysis determines the LRC contains 3106 atoms. When redundancy analysis is performed on the cluster, 200 of 274 H-bonds are determined to be redundant and 80 of 144 hydrophobic interactions are determined to be redundant, leading to a

redundancy score of 0.70. 3 of the H-Bonds and 4 of the hydrophobic interactions had criticality values of 0.29 or greater (see Table 6.5).

ID	Atom 1	Atom 2	Type	Energy	Crit. val.
1	26 GLU O	32 GLU H	HB	-6.98	0.29
2	40 ARG HH12	276 ARG OD2	HB	1.08	0.29
3	40 ARG HH22	276 ARG OD2	HB	-1.61	0.29
4	27 LYS CB	36 LYS CG	HP	N/A	0.29
5	27 LYS CB	36 LYS CD1	HP	N/A	0.29
6	36 TYR CE1	40 TYR CD	HP	N/A	0.29
7	36 TYR CZ	40 TYR CD	HP	N/A	0.29

Table 6.5: Critical interactions in the largest rigid cluster of DNA polymerase β (2FMQ). 3 H-bonds (HB) and 4 hydrophobic interactions (HP) with criticality values $\geq$0.25 were detected. The type, energy, and criticality value for each interaction are shown.

When any of these 7 interactions is removed, the lyase domain breaks off from the largest rigid cluster. The lyase domain does not form a single cluster, but instead shatters into many smaller clusters. For example, when we remove the H-bond between 26 GLU O and 32 GLU H, and rerun our redundancy analysis, the results are shown in 6.4e. The large rigid cluster remaining now contains no interactios with criticality value greater than 0.03. Results for removing any of the other 6 critical interactions are similar. The 2FMQ conformation is more closed than the 9ICI conformation. We repeated the redundancy analysis on 9ICI and found that the rigid clusters were in good agreement with the clusters of 2FMQ after the removal of any of the critical interactions (as in Figure 6.4e). For the largest rigid cluster of 9ICI, composed of 2336 atoms, the maximum criticality value of any of its consituent interactions was 0.03.

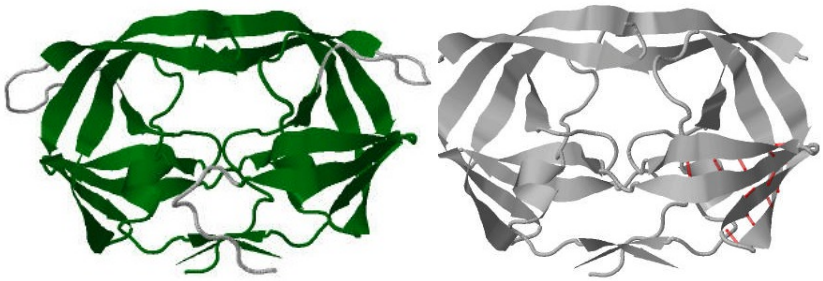
To summarize, the resulting decomposition determined by KINARI, run with default options, on the closed conformation (2FMQ) is very rigid, where the lyase

domain is included in a large rigid cluster spanning other functional domains of the protein. With our redundancy analysis, we found 7 interactions with very high criticality values (0.29 or greater). When any of these interactions were removed, the lyase domain decouples from the other domains and becomes very flexible, better matching the rigidity results for the open conformation (9ICI).

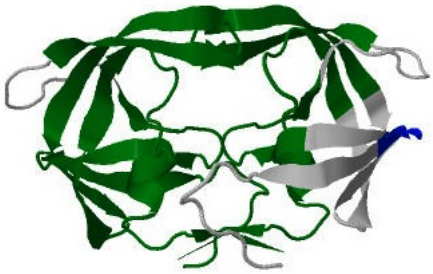
6.4.1.4 HIV-1 Protease

Our analysis of two conformations of HIV-1 Protease, 1HHP (open) and 1HTG (closed) uncovered interesting differences due to asymmetries in the 1HTG dimer. The 1HHP PDB file contains only a single chain, and therefore we computed the positions of atoms in chain B resulting in a dimer that is completely symmetric. The rigidity analysis results show a large rigid consisting of 78 of the 99 residues in each chain (residues 1-14, 19-36, 43-45, and 56-98). No interactions with criticality value greater than 0.10 were found.

For 1HTG, which was crystallized as a dimer, both chains are already included in the PDB file. The results of rigidity analysis on 1HTG reflect some of the asymmetries in the two chains (see Figure 6.5). With default options, the largest rigid cluster contains 81 residues from chain A (residues 9-33 and 43-98) and 90 residues from chain B (residues 1-33 and 43-98). More differences in the rigidity properties of the two chains are detected when analyzing the redundancy of the structure. 14 interactions were found criticality values $\geq$0.25 (see Table 6.6. All of these critical interactions were found in a β-sheet of chain A. Due to asymmetries in chain A and B, the set of H-bonds and hydrophobic interactions were not the same in the two chains. For example, in chain B, 65 GLU CG and 68 GLY C are a distance of 3.60Å and fell within the cutoff distance criteria for a hydrophobic interaction. The same pair of atoms in chain B were a distance of 5.18Å, much greater than the 3.65Å cutoff distance, and no hydrophobic interaction was placed.



(a) RCD, default (b) highly critical interactions



(c) RCD, without critical H-bond

Figure 6.5: Critical interactions in HIV-1 Protease (1HTG). 10 interactions with criticality values $\geq$0.10 were found for PDB 1HTG. (a) Before removing any interactions (b) Interactions with criticality values $\geq$0.10 were detected in the largest rigid cluster (LRC) are all H-bonds in the β-sheet of chain A. (c) After the removal of any of these interactions, the β-sheet of chain A breaks off from the LRC and becomes flexible.

6.4.2 Correlating redundancy and foldons, case study of Cytochrome-*c*

Figure 6.6a show the five largest rigid clusters of Cytochrome-c (1HRC). The rigidity of this protein has been previously investigated [78, 90, 94]. Here we perform a refinement of the analysis from a redundancy point of view (Table 6.7). When we focus on the largest rigid cluster (blue in Figure 6.1a and shown in Figure 6.1b) composed of two α-helices, we identify 24 H-bonds (shown in green) and 10 hydrophobic interactions (shown in blue). Each α-helix is held together by H-bonds, while the hydrophobic interactions effectively "zip-up" the two α-helices and hold them rigidly together. Redundancy analysis determines that 25% of the H-bonds and 40% of the 10 hydrophobic interactions are critical. Removing any of these critical interactions will cause the cluster to break up and become flexible. Most of these interactions

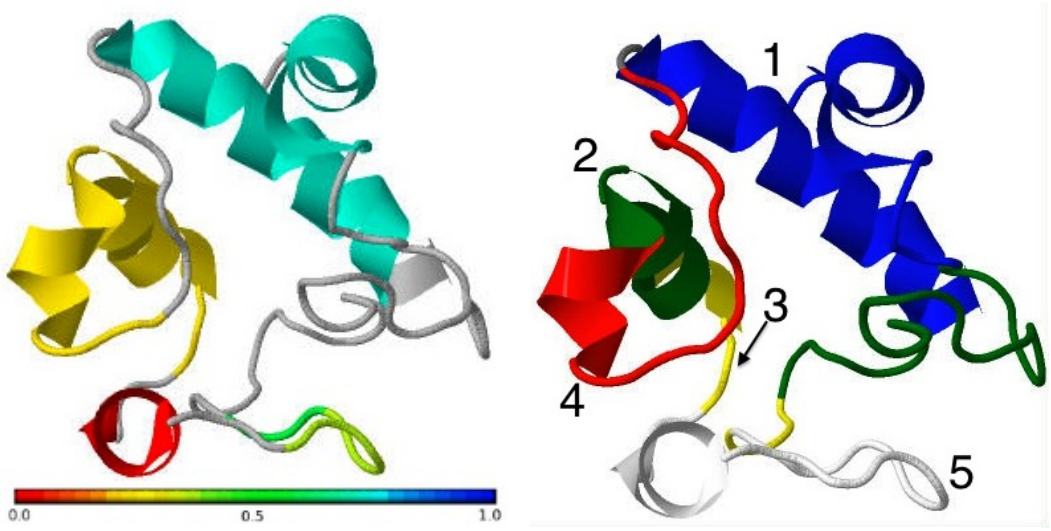
ID	Atom 1	Atom 2	Type	Energy	Crit. val.
1	9 PRO O	24 LEU H	HB	-7.22	0.36
2	11 VAL O	22 ALA H	HB	-5.82	0.36
3	11 VAL H	22 ALA O	HB	-5.53	0.36
4	13 ILE O	20 LYS H	HB	-5.98	0.27
5	13 ILE H	20 LYS O	HB	-5.13	0.27
6	14 LYS O	65 GLU H	HB	-6.17	0.27
7	14 LYS H	65 GLU O	HB	-3.34	0.27
8	15 ILE H	18 GLN O	HB	-6.89	0.27
9	64 ILE H	71 ALA O	HB	-6.28	0.27
10	62 ILE O	73 GLY H	HB	-4.10	0.27
11	62 ILE H	73 GLY O	HB	-5.52	0.27
12	64 ILE O	71 ALA H	HB	-5.57	0.27
13	66 ILE O	69 HIS H	HB	-0.82	0.27
14	66 ILE H	69 HIS O	HB	-4.94	0.27

Table 6.6: Critical interactions in the largest rigid cluster of HIV-1 Protease (1HTG). 14 H-bonds (HB) with criticality values ≥ 0.25 were detected, all in chain A. The type, energy, and criticality value for each interaction are shown.

Cluster	Atoms	HB redun / all	HP redun / all	Score	Max crit. val.
blue	251	18 / 24	6 / 10	0.73	0.44
left yellow	194	2 / 13	6 / 12	0.25	0.55
red	80	0 / 4	0 / 0	0.00	0.88
green	35	0 / 1	5 / 6	0.59	0.40
right yellow	30	0 / 1	2 / 3	0.36	0.34

Table 6.7: Redundancy scores for the five largest rigid clusters of Cytochrome c (1HRC), shown in Figure 6.6a. Listed for each cluster are number of atoms, numbers of redundant / all H-bonds and hydrophobic interactions within the cluster, our calculated redundancy score, and the greatest decrease in size observed after removing an interaction.

which we have labeled as 'critical' do not have a large impact on the cluster size. For each of the critical noncovalent interactions, we monitored how the original cluster



(a) (b)

Figure 6.6: Case study of Cytochrome-c (1HRC). We compare the rigid clusters and foldons. (a) The five largest rigid clusters of 1HRC are colored by their redundancy score, from least redundant (red) to most redundant (blue). (b) Experimentally determined foldons are numbered by their stepwise folding order

size of 251 atoms decreased when the critical interactions were removed. For all of the H-bonds, the cluster size after removal remained at least 86% of its original size. For three of the four hydrophobic interactions, the cluster size remained at least 98%. For one hydrophobic interaction, the cluster size dropped to 56%. Figure 6.1b shows how the cluster rigidity is affected when this particular interaction is removed. This demonstrates how each critical interaction may have a different degree of impact on a cluster's rigidity when they are removed.

Extensive studies have been undertaken to understand the folding kinetics of this protein. The foldons, intermediate structures which form during the folding process, and the order that they form, have been experimentally identified using HX experiments [67]. The foldons are blue (the 1-19 and 87-105, the N- and C-terminal α-helices), green (60-70, 19-36, α-helix and V-loop), yellow (37-39:58-61, short two-stranded antiparallel β-sheet), red (71-85 V-loop), and infrared (40-57 V-loop). Although the helix and V-loop with the green appear disconnected, the two parts engage

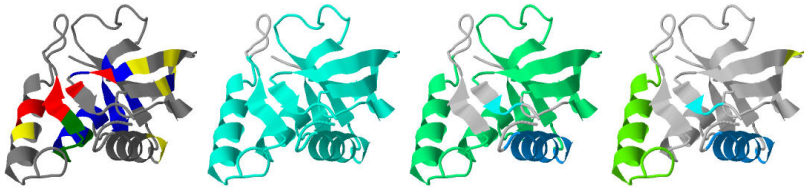
in hydrophobic interactions between sidechains PHE 36 and LEU 64. Not shown in the picture is the HEME-ligand.

Visually there is some nice agreement between these experimentally identified foldons and those determined by KINARI. In particular, the first foldon, the N- and C-terminal α-helices, correlate well with the most redundant rigid cluster found in our analysis. The α-helix of the 2nd foldon, and the entire 3rd and 4th foldons (yellow and red) lie in the yellow cluster, which has a lower redundancy. The last foldon, infrared, has not been placed into a single cluster but is instead determined to be flexible and lies in a number of clusters. A nice result is that the redundancy scores calculated (Table 6.7) also correlate with the foldon order.

Although this case study on Cytochrome-c demonstrates that KINARI with default parameters can assist in identifying these foldons, further analysis is needed to know if this extends to other proteins for which foldons are known. SNase is a 149-residue mixed α/β protein with three α-helices, a major five-stranded β-barrel, and three minor β-strands. It is composed of 5 foldons, the first foldon of which is composed primarily of the β-barrel. The default parameters for analysis of PDB 1SNP, SNase protein [7], reveal an almost entirely rigid structure, with no differentiation between the structural elements identified to form foldons. Excluding weaker H-bonds, the conventional parameter used to tune rigidity results, leads to the β-barrel losing rigidity before other foldon regions, which does not agree with the experimental data.

6.4.3 Survey on a Pdomain benchmark data set

We calculated critical and redundant interactions for the largest rigid cluster (LRC) of each protein in the Pdomain benchmark 3 data set (described earlier in Section 6.3.3). To get a better sense of how redundancy and the presence of critical interactions correlates with size, we divided up the 121 protein data set by the LRC



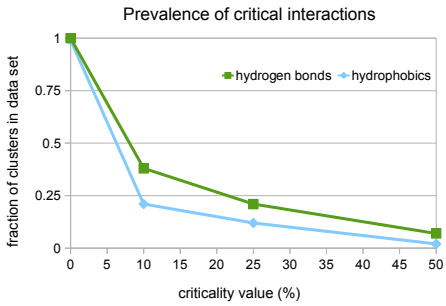
(a) foldons (b) default (c) -2 kcal/mol cutoff (d) -3 kcal/mol cutoff

Figure 6.7: Case study of SNase protein (1SNP). We compare the experimentally-determined foldons of SNase protein, 1SNP, with the KINARI rigid cluster decompositions. (a) Experimentally determined foldons. (b) With default options, the entire protein is determined to be almost complete rigid with a redundancy score of 0.72. (c) After removing H-bond using a cutoff of -2 kcal/mol, the minor β strands are determined to be flexible while the β barrel remains in the largest rigid cluster. (d) With a cutoff of -3 kcal/mol, the β barrel no longer lies in a larger rigid cluster, while two of the α-helices have remained rigid. Rigidity and redundancy analysis do not appear to give strong insight into this SNase foldons, unlike the Cytochrome-c, which is all alpha, for which there has been success in determining foldons.

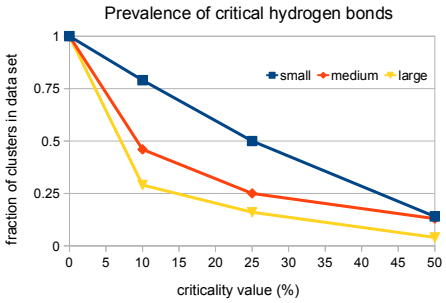
sizes: small (fewer than 500 atoms, 12% of data set), medium (500-1000 atoms, 20% of data set), and large (greater than 1000 atoms, 69% of data set).

Redundancy scores. Overall, the mean redundancy score was 0.74 (s=0.10). The small clusters had a lower mean redundancy, 0.66 (s=0.17), than the medium, 0.73 (s=0.11), and large, and 0.75 (s=0.07) clusters, showing a trend that the redundancy score increases and variance decreases with cluster size. Therefore, the larger rigid clusters are shown to be more robust, and less sensitive to changes in rigidity.

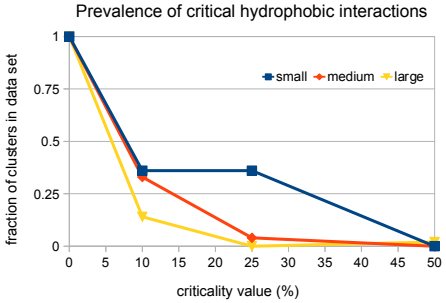
Prevalence or highly critical interactions. Figure 6.8 shows the cumulative distributions of clusters containing interactions with increasing criticality values. Although virtually all of the clusters contained some critical interactions, most of the clusters did not contain interactions with criticality value $\geq$0.10. Interactions with criticality values $\geq$0.50 were in 13% of the small and medium-sized clusters, but quite rare in the large clusters, occurring in fewer than 4% of them.



(a) overall



(b) H-bonds, by cluster size



(c) hydrophobics, by cluster size

Figure 6.8: Prevalence of critical interactions in Pdomain benchmark data set. (a) Overall comparison of occurrence of critical H-bonds and hydrophobic interactions. (b-c) Occurrence of critical H-bonds and hydrophobic interactions for small, medium, and large-size clusters.

6.4.4 Comparison with other techniques

MSU-FIRST included a feature to assign each bond a *flexibility index*, using a count of the redundant constraints [48]. In the underlying bar-and-joint model of the protein used by MSU-FIRST, each bond is represented by a number of constraints: *central-force constraints* for holding bond lengths, and *external constraints* for holding bond-bending and dihedral angles. Using the MSU-FIRST pebble game, the input macromolecule is decomposed into three types of regions: *isostatically rigid*, *overconstrained*, and *underconstrained*. An isostatically rigid region is a rigid cluster in which the removal of any of the constraints will cause the cluster to become flexible. An overconstrained region is a rigid cluster in which at least one of the constraints

is redundant. An underconstrained region is a region in which placing any additional bond-bending or dihedral angle constraints will cause a rigid cluster to form.

Once the different regions were identified, Equation 6.2 was used to calculate the flexibility index for each bond [48]. H_k and F_k are the number of rotatable bonds and the number of degrees of freedom in the kth underconstrained region. C_j and R_j are the numbers of bonds and redundant constraints within the jth overconstrained region. The flexibility index is negative for overconstrained regions and positive for underconstrained regions.

$$f_i \equiv \begin{cases} \frac{F_k}{H_k} & \text{in an underconstrained region} \\ 0 & \text{in an isostatically rigid region} \\ \frac{-R_j}{C_j} & \text{in an overconstrained region} \end{cases} \tag{6.2}$$

The flexibility index was defined for bonds within any region in the protein, not just the rigid clusters. In order to directly compare our redundancy score, we transform the flexibility index formula to an equation for scoring rigid clusters (Equation 6.3). $\Psi(i)$ is the MSU-FIRST redundancy score for a cluster i, where R_i and C_i are the numbers of redundant and central-force constraints within the cluster.

$$\Psi(i) = \frac{R_i}{C_i} \tag{6.3}$$

This scoring formula and ours (Equation 6.1) are not equivalent. We demonstrate this with a small example. A cluster consisting of a ring of 5 atoms connected by 4 single covalent bonds and one H-bond will be assigned a score of 0 with our method. The score signifies that there is no redundancy in the set of noncovalent interactions, and if the H-bond were removed, the cluster would break. MSU-FIRST assigns the same cluster a score of 1/5 (1 redundant constraint and 5 bonds). Unlike our approach, the MSU-FIRST approach does not detect the critical interactions.

ASU-FIRST also included a flexibility index which does not use any information on redundancy. The index for bonds in rigid clusters is based on the size of the cluster [88].

6.4.5 Further directions

We proposed a method for understanding which interactions are critical in maintaining a protein's 3D shape. This may be applicable to other related questions previously posed in the literature.

Energy. Energy functions provide a way to compare the relative strengths of H-bonds. Different functions have been proposed that calculate the energy required to break a H-bond using the local bond geometry [58,68]. We may infer that the stronger interactions, the ones that require more energy to break, are the critical ones.

Flickering. The *flickering* phenomenon is the forming and breaking of interactions at varying rates during the natural fluctuations of a protein about the native state [61]. The *duty cycle*, which is the percentage of time a particular interaction is present, may be used similarly to energy, to rank interactions by how likely they are to break. An advantage of the duty cycle concept is that it generalizes to any interaction which may break and form. This is especially valuable for hydrophobic interactions since the associated energy is unknown.

Evolutionary conservation. Using proteins within the same family with high structural conservation, it has been shown that even when there is low-sequence identity, a network of hydrophobic interactions between residues are conserved [36]. The conserved interactions may be considered the "critical" ones, and similar to the high duty cycle interactions, identifying conserved interactions does not rely on computing an energy function.

Other flexibility index methods. Another type of flexibility index was computed on the amino acid sequence using parameters derived from B-values from a training

set of PDB files [53, 91]. Yet another method used normal mode analysis to calculate the local chain deformability for each residue along the backbone [59]. This method was shown to produce comparable resutls to the MSU-FIRST flexibility index in a case study on 16pk kinase [59].

Dilution. Dilution reveals an unfolding pathway of a protein by removing H-bonds one-by-one and performing rigidity analysis. It would be interesting to correlate the set of H-bonds identified as critical using our method, with those which cause the greatest changes in the rigidity during dilution. Dilution was used as a tool to show the coordinated states of thermophilic and mesophilic protein homologues [33]. This study used measures of the rigidity properties to identify the transition point from flexible to rigid, and showed that in 2/3rds of the proteins in their data set, the transition points occurred at higher temperatures in the thermophilic than the mesophilic homologues. The classification of the critical and redundant interactions may be used as additional information to improve the order in which the bonds are removed in dilution, which may lead to more consistently corresponding transition points.

Computational efficiency. The algorithm for classifying all interactions as critical or redundant, described in Section 6.3.1, takes in worst case cubic time in the number of atoms, so the method does not scale well to proteins with more than 500 residues. We have shown that due to the rarity of very critical interactions, a uniform sampling approach is inadequate. Because these critical interactions did tend to be concentrated together, a targeted sampling approach may be sufficient if some knowledge of the structure is known a priori. Another reasonable approach to speed up the algorithm would be to devise a method that leverages common intermediate states of the pebble game, so that a new run of the pebble game would not need to be peformed for each interaction.

6.5 Conclusion

Motivated by a need to understand the sensitivity of rigid clusters to changes in the set of noncovalent interactions, we proposed a method for classifying the noncovalent interactions as *critical* or *redundant*. An interaction is *critical* if, when it is removed, the cluster it is contained in breaks up and becomes flexible . We also proposed a method to score clusters using the redundancy of the noncovalent interactions. We have implemented these methods as extensions to KINARI, our protein rigidity analysis software. We provide results of our classification and scoring for the clusters of a small data set of PDB files.

CHAPTER 7

EXTENSIONS

In this chapter, we propose future work which builds upon the contributions of this thesis. We begin with potential extensions to the KINARI software, presented in Chapter 3, to expand the classes of mechanical models that can be processed via the available rigidity analysis algorithms. In particular, we discuss our ideas for processing non-generic mechanical models; this is new work and to the best of our knowledge, these problems have not been posed elsewhere. Later in this chapter, we discuss extensions to the benchmarking methodology presented in Chapter 4. We also propose extensions to the methods for modeling noncovalent interactions, building on work presented in Chapter 5. Finally, we discuss how the redundancy analysis work, presented in Chapter 6, should be extended.

7.1 Applying rigidity analysis to a larger family of mechanical frameworks

The KINARI software is the first to provide access to a mechanically accurate underlying model. Tay's theorem is only guaranteed on generic frameworks, and in order to provide the user greater control in modeling choices, we permit non-generic options (such as multi-bar modeling) and offer heuristics, as discussed in Sections 3.2.2 and Section 5.3.2.

The KINARI software supports non-generic modeling. For example, the code supports representing multi hinges, which connect more than 2 bodies. We have also

provided functions in the KINARI kernel library to confirm whether a body-bar-hinge framework is 'combinatorially' generic.

For future work, the KINARI kernel library should be extended to support mixed structures, called atom-body structures, which are more general than body-bar-hinge frameworks. We define concepts for describing atom-body structures, and then use these concepts in our proposed modules for detecting whether the structure is a combinatorially generic body-bar-hinge framework.

7.1.0.1 Atom-Body Structures and Tay's Theorem

A hypergraph $H = (V, E)$ is composed of a set of vertices and a set of hyper edges (or subsets) of vertices. A *geometric hypergraph*, (H, p) is a hypergraph and a function $p : V \mapsto \Re^d$ which assigns coordinates to all of the vertices.

We are interested in a particular class of geometric hypergraphs, which we call an *atom-body structure*, for which the following three properties hold:

1. *Covering.* Every vertex is contained in some subset.

2. *Non-inclusion.* No set is a subset of any other set.

3. *Bounded intersection.* The intersection of any two subsets has size at most $d-1$, where d is the dimension.

For the remainder of this chapter, we work with atom-body structures in dimension 3. When testing for the bounded intersection property, we hold $d = 3$.

We call a subset of size 1 an *atom*; a subset of size 2 a *bar*; a subset of size 3 a *panel*; and a subset of size 3 or more, a *body*.

For helping in the succinctness of our definitions, we denote $E(v)$ as the set of hyper edges or subsets, $D(v)$ as the set of bodies, $R(v)$ as the set of bars, and $A(v)$ as the set of atoms that a vertex v is contained in.

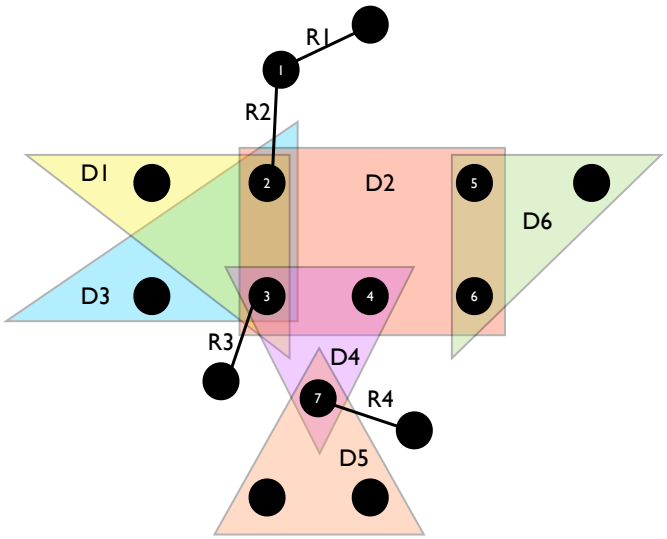


Figure 7.1: Example atom-body structure. The 7 joint points are numbered.

Joints. For a vertex v, if v is in more than one subset, $|E(v)| > 1$, then v is a *joint*.

Pins. For a joint v, where the subsets of v are $E(v) = e_1, e_2, .., e_k$, if $|e_1 \cap e_2 \cap ... \cap e_k| = 1$, then v is a *pin joint*. If v is a pin joint, if $|E(v)| = 2$, v is a *simple pin*. Otherwise v is a *multi pin*.

Hinges. For every pair of joints v_1 and v_2, if $D(v1) \cap D(v_2)$ is non-empty, then v_1 and v_2 form a *hinge*. We call the two joints in the hinge, *hinge joints*. If $|D(v1) \cap D(v_2)| = 2$, the hinge is a *simple hinge*. Otherwise, the hinge is a *multi hinge*.

Hypergraph duals Given a hypergraph $H = (V, E)$, the dual $H* = (V*, E*)$, every edge $e_i \in E$ becomes a vertex v_i* and every vertex $v_j \in V$ becomes an edge $e_j* = v_1*, ..., v_k*$ where $e_j* = v_i * | v_j \in e_i$. The dual for the example atom-body structure is shown in Table 7.1.

Joint features. There are three types of joints:

1. all-bar joint

2. all-body joint

Joint	Vertex	Bars	Bodies
1	1	R_1 R_2	
2	2	R_2	D_1 D_2 D_3
3	3	R_3	D_1 D_2 D_3 D_4
4	4		D_2 D_4
5	5		D_2 D_6
6	6		D_2 D_6
7	7	R_4	D_4 D_5

Table 7.1: Dual of the example hypergraph shown in Figure 7.1. For each vertex, the hyper edge in the dual is composed of the collection of atoms, bars, and bodies that the vertex lies in. In this table, we only include the vertices that occur in more than one body (the joints). Our example shows no subsets that are atoms, so we exclude this in the table.

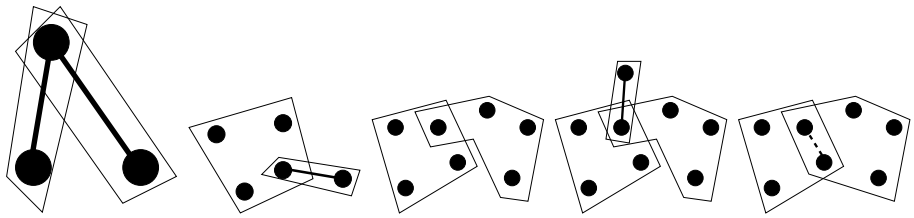


Figure 7.2: Examples of joints. (a) all-bar simple pin joint, (b) bar-body simple pin joint, (c) all-body simple pin joint, (d) bar-body combination multi pin joint, (e) two hinge joints forming a simple hinge.

3. bar-body combination joint

Figure 7.2 shows examples of joints.

In Table 7.2, we show the intersections of the hyperedges of the dual. We only show the intersections of the bodies, and not the bars. For any two joints, if the size of the intersection of the bodies of the joints, $|D(j_1) \cap D(j_2)| \geq 2$, then the two joints form a hinge. If $|D(j_1) \cap D(j_2)| = 2$, the hinge is a simple hinge. If $|D(j_1) \cap D(j_2)| \geq 3$, the hinge is a multi hinge.

J_a	J_b	$D(J_a) \cup D(J_b)$
2	3	$D_1\ D_2\ D_3$
2	4	D_2
2	5	D_2
2	6	D_2
2	7	
3	4	$D_2\ D_4$
3	5	D_2
3	6	D_2
3	7	D_4
4	5	D_2
4	6	D_2
4	7	D_4
5	6	$D_2\ D_6$
5	7	
6	7	

Table 7.2: The intersection of the hyper edges of the dual helps us to identify hinges. We have excluded joint 1 and bars in this table.

Isolated hinges. When every body that contains a joint in a hinge also contains the other joint in the hinge, we call this an *isolated hinge*. For an isolated hinge composed of joints v_1 and v_2, $|D(p_1)| = |D(p_2)| = |D(p_1) \cap D(p_2)|$.

Hinge incidences. When a joint in a hinge is also a bar-body joint, we call this a *bar-hinge concurrency*. When two or more hinges share a joint in common, we call these hinges *concurrent hinges*. See Figure 7.4.

We now describe a special type of all-body joint which are prominent in models of molecules. When a joint is in multiple hinges, but all the hinges are common to one body, we call this joint a *molecular joint*. Figure 7.6 show examples of joints that are not molecular joints.

Tay-combinatorially-generic atom-body structures. Our goal in classifying these features of atom-body structures is to:

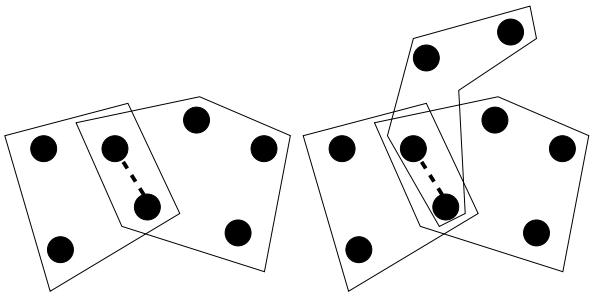


Figure 7.3: Example hinges. A simple hinge (a) and a multi hinge (b).

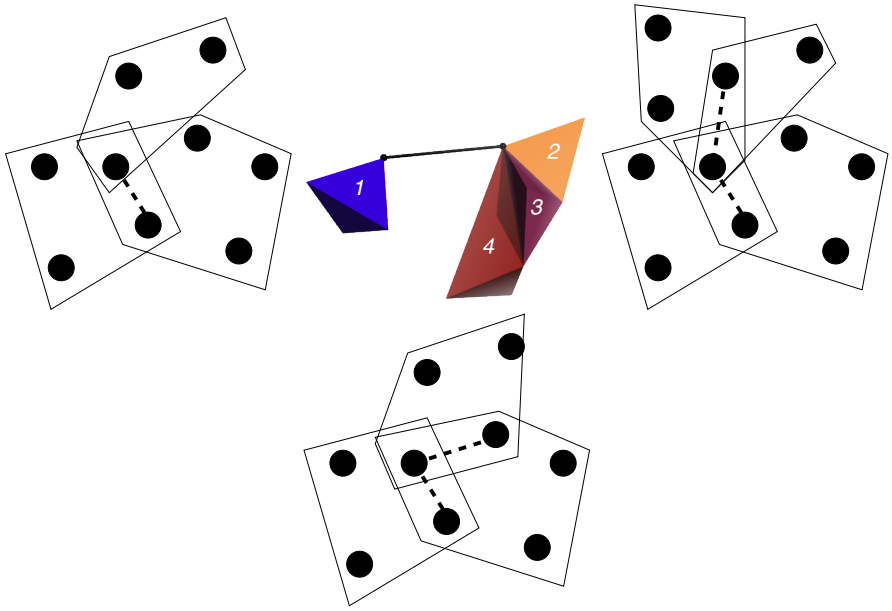


Figure 7.4: Hinge incidences. (a) a hinge that contains a multi-joint, (b) a bar-hinge concurrency. (c) and (d) concurrent hinges.

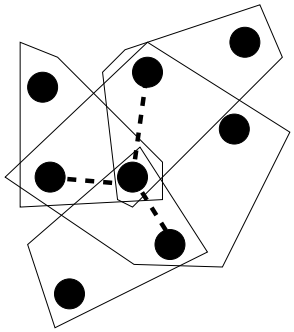


Figure 7.5: Molecular joint. The joint shared by the concurrent hinges is a molecular joint because there is one body to which all the hinge joints belong.

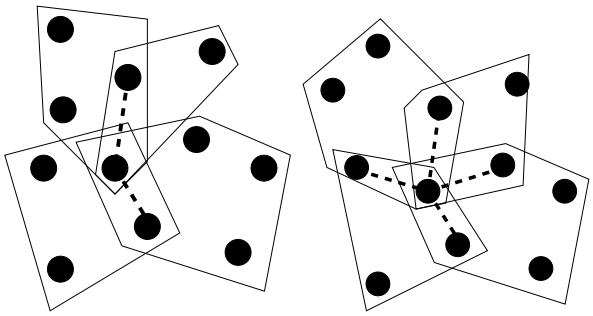


Figure 7.6: Non-molecular joint. The joints shared by the concurrent hinges are not molecular joints because the joints in the hinges do not all lie in the same body.

1. decide if an atom-body structure is valid to analyze using Tay's theorem, assuming all points lie in generic position

2. if it is not valid, then to identify the degenerate combinatorial features

Definition 1. ***Without the molecular conjecture****, an atom-body structure is* ***Tay-combinatorially-generic*** *if the following conditions are satisfied:*

- ***Proper.*** *The "Covering", "Non-inclusion", and "Bounded intersection" properties hold.*
- ***No singletons.*** *There are no atoms.*
- ***No dangling bars.*** *Each bar endpoint is also a joint.*
- ***Proper joints (without molecular conjecture).*** *Every joint is either a bar-body simple pin or an isolated simple hinge joint.*

The above conditions to be Tay-combinatorially-generic exactly align with the description of a body-bar-hinge framework in which there are no multi hinges, no concurrent hinges, no concurrent bars, and no bar-hinge concurrencies. Tay's theorem is guaranteed to work on such frameworks. We know, because of the molecular conjecture, that Tay's theorem is also guaranteed on some body-bar-hinge frameworks with concurrent hinges. We expand the conditions for an atom-body structure to be

considered "Tay-combinatorially-generic", and then make a conjecture that Tay's theorem is also guaranteed for such atom-body structures.

Definition 2. ***With the molecular conjecture****, an atom-body structure is* ***KT-generic*** *if the following conditions are satisfied:*

- ***Proper.*** *The "Covering", "Non-inclusion", and "Bounded intersection" properties hold.*
- ***No singletons.*** *There are no atoms.*
- ***No dangling bars.*** *Each bar endpoint is also a joint.*
- ***Proper joints (with molecular conjecture).*** *Every joint is either a bar-body simple pin or a molecular joint.*
- ***No multihinges.*** *There are no multihinges.*

Problem statement. Given an arbitrary atom-body structure, we are interested in answering the following questions:

1. In its current form, is it Tay-combinatorially-generic (or KT-generic)?
2. If it is not Tay- (or KT-) generic, what are the features which make it non-generic?
3. Is there an equivalent form that we may transform it to (for example, by removing bars in which both endpoints are contained in the same body) so that it is Tay- (or KT-) generic?

7.1.0.2 Identifying combinatorial degeneracies

We present a method for processing an atom-body structure to identify some of the degeneracies that the user may consider and possibly fix. We describe first a set of modules then the process for applying them.

- *Module 1. Check covering property.* Detect whether every point is in some atom, bar, or body.

  Input An atom-body structure

  Output The points in the atom-body structure which are not found in any atom, bar, or body.

- *Module 2a. Check non-inclusion property.*

  Input An atom-body structure

  Output All atoms, bars, and bodies which may be discarded because they are contained in another atom, bar, or body.

- *Module 2b. Check bounded-intersection property.* Identify all bodies which could be merged.

  Input A set of bodies.

  Output Subsets of bodies which overlap on three or more points and should be merged. (See algorithm below)

- *Module 3. Check no-singleton property.*

  Input An atom-body structure

  Output All atoms (subsets of size 1).

- *Module 4. Detect if points in bars are contained in some body.*

  Input A set of bars and bodies.

  Output All points which lie in a bar but not a body.

- *Module 6. Detect pins.*

  Input. A set of bars and bodies.

  Output. All pin.

- *Module 8. Detect bar concurrencies* If a bar lies in a connected component of size 2 or more, then it is bar-concurrent.

Input. A set of bars.

Output. Connected components of bars.

- *Module 7. Detect hinges.*

  Input. A set of bars and bodies.

  Output. All hinges. (Each hinge contains information about the incident bodies).

- *Module 7b. Detect isolated simple hinges*

  Input. A set of hinges, bars

  Output. All isolated simple hinges.

- *Module 9. Detect concurrent hinges.* Every connected component of hinges of size 2 or more contains concurrent hinges.

  Input. A set of hinges.

  Output. All concurrent hinges.

- *Module 10. Detect bar-hinge incidences.*

  Input. A set of bars and hinges.

  Output. All bar-hinge incidences. (Find connected components of bars and hinges. If a bar and a hinge lie in a connected component, and are incident, this is a bar-hinge incidences.)

We describe the steps for identifying if an atom-body structureis Tay-combinatorially-generic.

- Perform module 1 and check if all points lie in some subset. If module 1 does not hold, the user must deal with this.

- Perform module 2a and 2b, and check if inclusion property holds. Attempt to "absorb" contained sets and glue bodies, and check again.

- Perform module 3, and check if there are any atoms. If so, user intervention is required. The user may choose to hand-edit the data to remove them.
- Perform module 4, and check that every point in a bar also lies in a body. If not, user user intervention is again required to transform the data.
- Perform module 5, and then merge all bodies which are found to be rigid by the gluing-lemma.
- Perform module 6. If no pins found, continue. Else, return the error to the user.
- Perform module 7. If all hinges found are simple hinges, continue. Else return the error to the user.
- Perform module 8-10. If any bar / hinge incidences found , return the error to the user.

If the atom-body structure passes these steps with no errors, then we know that it is a Tay-combinatorially-generic body-bar-hinge and that Tay's theorem is guaranteed.

Algorithm 6 Computing all maximal bodies using gluing lemma.

Input. Sets of bodies.
Output. Sets of bodies that have been merged.

Build an adjacency list with weights. Each body is a vertex. The weight for each edge is the number of points that the two bodies share.
while There exists an edge with weight 3 or more in the adjacency list. **do**
Contract the edge (uv) by merging u and v, and recomputing the weight of the edges that were incident to u and v.
end while
Convert the graph back to bodies.

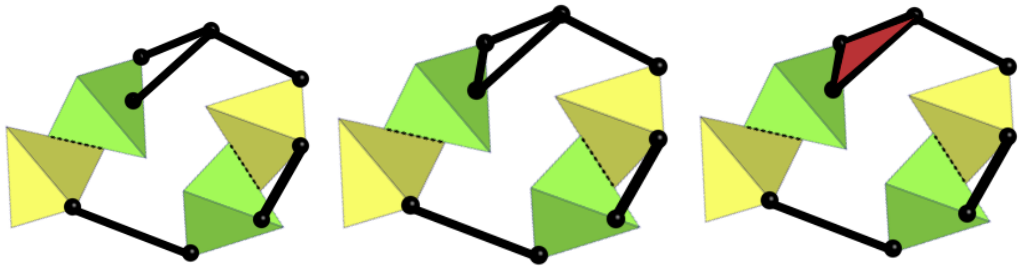


Figure 7.7: Example application of Module 11. The framework in (a) is nongeneric. More than one bar is attached at a joint, and the joint is isolated– it is not attached to a body. We observe that for two of these bars, the endpoints are attached to the same body and therefore rigidly attached to eachother. This is equivalent to a bar between the two endpoints, as shown in (b). A triangle bar framework is rigid in any dimension. We can relace these three bars with a triangle panel (or body) and now the framework is a generic body-bar-hinge (c).

7.1.0.3 Repairing and reducing combinatorial degeneracies

Each connected component of two or more bars detected in module 4 presents some obstacles in building a Tay-graph. The number of bars may be reduced by identifying additional bodies created by the bars, and then removing the bars associated with them. The connected components may be viewed as 3D bar-and-joint frameworks, so identifying the rigid bodies of the atoms within such frameworks is non-trivial. There is no known polynomial-time algorithm for doing such.

We identify two features of bars that may be converted into bodies.

- *Triangle body.* When two bars attached to one body share an end-point, as shown in Figure 7.7.

- *Tetrahedral extension.* When three or more bars attached to one body share an end-point. See Figure 7.8.

We extend the set of bars in a connected component by placing an additional bar between a pair of end-points if they lie in the same body. Adding these extra bars may only increase the number of rigid bodies found in the bar-and-joint framework

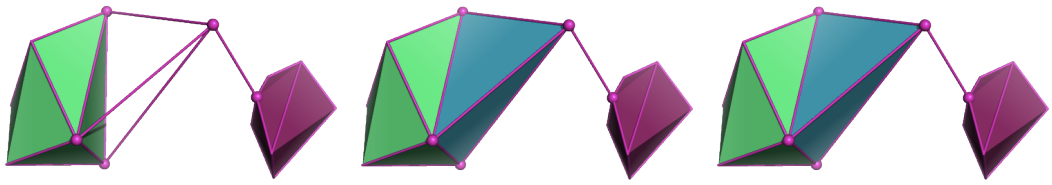


Figure 7.8: Example application of Module 12. The framework in (a) is nongeneric. We observe that the three bars extending from the left body form a rigid tetrahedron (b). The framework can be transformed to a generic one by extending the body on the left to contain this point. The convex hull is shown (c).

made from the connected component. We use a clique-finding algorithm to identify all triangles and tetrahedra and replace the associated bars with bodies.

- *Module 11: Finding additional triangle bodies in set of bars.*

  Input. A connected component of bars.

  Output. All triangles that form rigid bodies.

- *Module 12: Finding additional tetrahedral bodies in set of bars.*

  Input. A connected component of bars.

  Output. All tetrahedra (K4s) that form rigid bodies.

After running modules 11 and 12, module 2 may be applied to merge bodies.

In addition to these two examples which are found by finding some patterns in the bars, some degeneracies may be "fixed" by using the pebble game as a submodule. We observe that adding extra constraints, such as pins or bars, can only further rigidify a set of bodies. If a body-bar-hinge framework is rigid without adding the contraints that cause degeneracies, this implies that the original body-bar-hinge framework is rigid with the degenerate constraints. We suggest this as an additional method, which we may investigate further later.

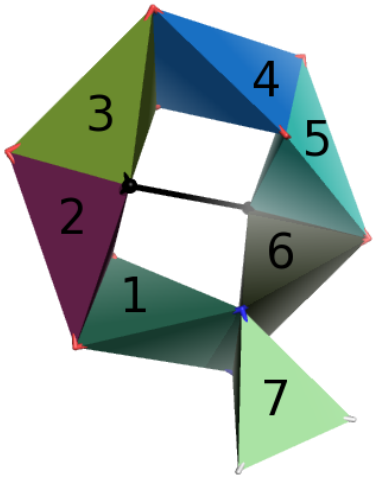


Figure 7.9: An example of a body-bar-hinge framework for which heuristic techniques for rigidity analysis may apply.

7.1.0.4 Rigidity analysis on non-generic frameworks

For frameworks which still contain degeneracies after performing triangle and tetrahedral transformations, other approaches may be proposed for determining the rigidity. For example, Figure 7.9 shows mechanical model composed of seven bodies, a bar, and seven hinges. A multi hinge exists between bodies 1, 6, and 7. The bar endpoints align with hinges. This is the type of structure which cannot be processed with current techniques, because it does not have a defined Tay graph. A decomposition plan, to first analyze subcomponents of the framework which are generic, would permit the rigidity of the framework to be analyzed.

7.2 Extending benchmarking of rigidity analysis systems

In Chapter 4, we proposed a methodology for benchmarking rigidity analysis systems. This was based on computing a similarity score of predicted rigid cluster decompositions and those from a 'gold standard' data set. In our evaluation, we used the cluster decompositions and data set of the Gerstein Lab's RigidFinder server.

This benchmarking methodology was designed to rapidly evaluate whether a cluster decomposition has high overlap with the gold standard, and assisting in finding the optimal setting for tuning parameters. This method is not especially sensitive to detecting flexibility in small loop regions. This issue might be partially attributable to the benchmarking data available. Ideally, scores from multiple different benchmarks would be combined together in order to rank different cluster decompositions methods. One resource that could be leveraged is the Gerstein Lab's hand-annotated hinge listing that was used to validate a number of the lab's prediction methods [20, 56].

Methods to compare against. We compared with very crude baselines, the all-floppy and all-rigid baselines. Future studies should compare with other decomposition methods, such as those based on sequence analysis [17, 20] and energy functions [19].

7.3 Improving modeling accuracy

Varying the number of energy ranges and using multi-bar modeling In this thesis, we have presented the first study to evaluate the effects of varying the modeling the hydrogen bonds by energy. We also studied the effects of varying both the hydrogen bond energy cutoff and hydrophobic interactions. There were many other factors that we might have varied.

- *Multi-bar modeling.* We looked only into modeling with hinges or 1 bar, but our software also supports modeling a constraint as 2 to 6 bars. Future studies might include these other options in the evaluation to measure if accuracy increases.

- > 2 *energy ranges.* For the hydrogen bonds, we limited our study to two energy ranges: strong and weak. We could have an arbitrary number of energy ranges.

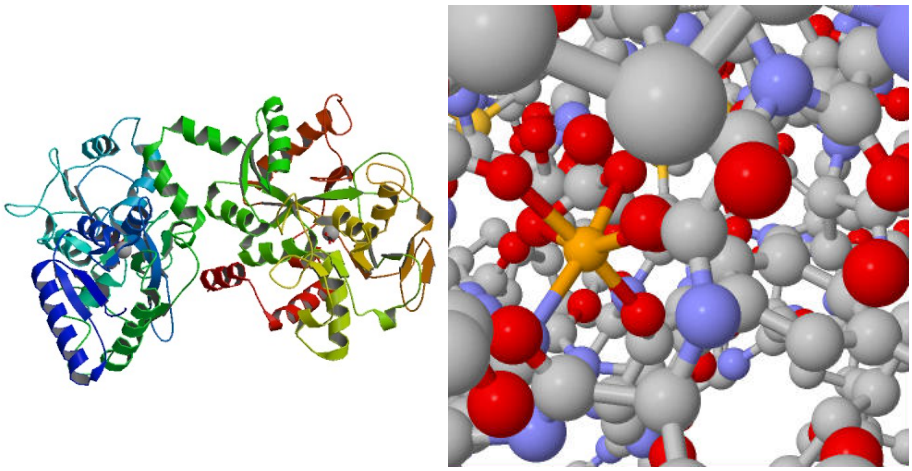
Extending types of supported interactions

There are a number of other stabilizing reactions that we have not included in our evaluation. First, by default in KINARI v1.0, all water molecules are removed. Hydrogen bonding with water molecules can contribute to the stability of the molecule, and therefore, a very accruate system would incorporate these interactions and reflect their contribution.

Ionic interactions, or "metal bonds", are of particular importance because the presence of a metal ion is crucial to the conformation or activity of over one third of all proteins [82], and a recent study published in Nature reveals that metal-binding proteins are even more abundant, than previously thought [14]. There is a push in the protein research community to better understand how these interactions with metal ions affect protein function. For example, Figure 7.10 shows Lactoferrin, a homodimeric iron-binding protein which transports irons to cells and regulate the level of iron in the blood. An iron ion binds to the protein with metal/ionic bonds. Another interaction type which has not been studied in the context of rigidity analysis are electrostatic interactions. These have also been shown to play a structural role in proteins, especially in protein-protein interfaces within complexes [72].

How to set parameters to apply rigidity analysis to a protein. The benchmarking methodology presented in Chapter 4 and further applied in Chapter 5 serves to find the best parameter setting during an evaluation when the gold standard is known previously. Our results confirmed that there is no universal parameter setting that gives optimal results for all proteins, but the results can be used to guide the choice of which parameter settings to use.

This data may be leveraged in the use of rigidity analysis as a predictive tool. Future studies should examine if an approach using training data may guide setting parameters.



(a) (b)

Figure 7.10: Example of metal-binding interactions in proteins. The protein shown is Lactoferrin (1B0L). (a) Folded structure with iron atoms shown in grey. (b) The two iron atoms (orange), each coordinated with 6 atoms, help to stabilize the structure.

7.4 Characterizing robustness of rigidity results

The redundancy analysis we proposed in Chapter 6 adds additional value to the RCD produced in a single rigidity analysis, for instance the results of KINARI-Web v1.0 (see Chapter 3, Section 3.3.3). The redundancy score of each cluster gives some sense of confidence of a cluster's stability.

Fuzzy rigid clusters. In our analysis, we kept track of the decrease in size of the rigid cluster as interactions were removed, but we did not make use of the information of which regions within the cluster more frequently break off than others. We hypothesize, with data supported by dilution experiments [40], that each cluster has a core, which is probably always retained by the cluster. Future work would validate whether this core correlates with the folding core determined in dilution. Additional value to KINARI-Redundancy results would be added by coloring atoms in a cluster based on how likely they are to break off in a redundancy analysis. This data might be compared with the 'fuzzy domains' calculated by another method, which relied on normal mode analysis [98]

Effects of removing multiple interactions. We demonstrated that there exist single interactions, that when removed, can have a big impact on the rigidity of a cluster. An interesting extension of this approach would be to study the effects of removing multiple interactions at a time. It may be the case that an interaction by itself is not especially critical, but perhaps it has some partner interaction, and when both of these are removed, the effects are detrimental to the rigidity of the cluster. Performing such studies exhaustively will quickly become computationally intractable. For example, KINARI v1.0 curation computes approximately 250 noncovalent interactions for the closed conformation of HIV-1 protease (1HVR). Extending our approach to examine the contribution of each pair of interactions would require ($\binom{250}{2} = 31,125$) invocations of rigidity analysis. Further extending to check every 3 interactions requires 3 million invocations. Therefore, the approach is not scalable in this respect, and more efficient methods would need to be developed.

Sampling interactions. Recent work in the Jacobs lab proposes heuristics to extend rigidity results to approximate ensemble-averaged rigidity [35]. With this method, each noncovalent interaction is assigned some probability. These probabilities could be used to repeatedly sample from the set of interactions, and perform rigidity analysis. Then the RCDs produced would be combined into a single RCD with some weighting scheme. No study has yet been performed to validate whether the ensemble-averaged rigidity better correlates with biological data than a single rigidity analysis.

CHAPTER 8

CONCLUSIONS

The goal of this thesis is to make progress toward accurate and robust software for protein rigidity analysis. The contributions of the thesis are summarized in the proceeding sections, but first, we provide a list of highlights:

- **Chapter 3.** We presented KINARI, an extensible and publicly-available software library for mechanical modeling and rigidity analysis, with applications in molecular modeling (`http://kinari.cs.umass.edu`).

- **Chapter 4.** We proposed a new benchmarking methodology which can be applied to any system for decomposing a protein into rigid and flexible regions. The approach comparatively scores two cluster decompositions of the same protein by applying the B-cubed scoring method from the information retrieval literature.

- **Chapter 5.** We proposed a new methodology for modeling hydrogen bonds by their energy. Weak hydrogen bonds are modeled with a weaker constraint, rather than being excluded completely as was done in prior work.

- **Chapter 5.** We performed a study to measure improvement in accuracy by tuning the energy cutoff for hydrogen bonds and hydrophobic interactions.

- **Chapter 6.** We present results from our exhaustive study of sensitivity of rigid clusters to variations in noncovalent interaction network on a benchmark dataset.

8.1 Summary of Contributions

A protein's characteristic rigidity and flexibility is essential to its function. Protein rigidity analysis systems process structural data, determined by laboratory experimental methods, and provide a course-grained mechanical model, which describes the rigid and flexible regions. A huge advantage of rigidity analysis over other computational methods is its computational efficiency. A rigid cluster decomposition of a 100 residue protein is determined in seconds.

Prior work with the MSU-FIRST and ASU-FIRST software demonstrated the power of rigidity analysis for testing real biological hypotheses. These software systems served as powerful tools to demonstrate the usefulness of rigidity analysis in a number of studies, but ongoing progress was stunted because the systems were not built modularly, to permit customization of the modeling. Perhaps for this reason, evaluation of the predictive power of the systems for performing rigid cluster decomposition was limited to very few case studies. Qualitative case studies should not be under-appreciated as contributions; they serve as powerful tools for demonstrating the usefulness of a system. But this approach to validation needs to be complimented with an unbiased, quantitative evaluation.

8.1.1 KINARI Software.

Towards our goal of validating the predictive power of rigidity analysis, we have designed and written the KINARI software as a platform for rigidity analysis experiments. KINARI is the first software to implement correct mechanical modeling of molecules as body-bar-hinge frameworks. The analysis of such frameworks is well-supported by mathematical theory and efficient, precise algorithms. In this thesis, we described the important concepts in KINARI to provide full reproducibility of our results. These included curation of the protein structural data, from PDB files, to a molecular representation, and then the modeling algorithm, to convert the molecular

representation into a mechanical structure. The other contributions of this thesis were performed with extensions of the KINARI software.

8.1.2 Towards improving accuracy of protein rigidity analysis systems.

In order to make progress in any research area, benchmarks must be in place so that improvements can be measured quantitatively. We have proposed a benchmarking methodology for measuring the accuracy of rigid cluster decomposition predictions. Our method borrows from the information retrieval literature, employing the B-cubed method for comparing two clusterings of data [2].

Our benchmarking methodology made it possible for us to evaluate our new methods for modeling noncovalent interactions. These are the interactions which are mainly responsible for holding a protein in its 3D folded shape. How they are incorporated into the mechanical modeling is vital to the accuracy of the results. In prior work, weaker hydrogen bonds were removed via an energy cutoff. Yet the contribution of these weaker hydrogen bonds to a protein's stability is non-negligible. We proposed a new methodology for their inclusion, in which, rather than removing the weaker hydrogen bonds entirely, they are simply modeled with a weaker constraint.

Although others have observed earlier that a balance between hydrogen bonds and hydrophobics was essential to achieving proper rigidity results [32], the effects of tuning the set of hydrophobics had never been thoroughly evaluated. Hydrophobics were not originally included in the MSU-FIRST software, and in the ASU-FIRST software, were determined with heuristics with no associated energy. We proposed to instead use van der Waals interactions, which can be calculated with a molecular mechanics package (we implemented the functions of Amber99 [8]), and assign energies based on their Lennard Jones 6-12 potentials.

Our evaluation using the aforementioned benchmarking methodology validated that KINARI v1.0, with default options, performed significantly better at determining

RCDs than our crude baselines on the larger proteins in our data set. But we also found that KINARI v1.0 performed worse than the baselines for the medium and small-sized proteins in the data set. The best overall performance could be attained by varying both the hydrogen bonds and hydrophobic interactions, thus finding the balance between the two interaction types is essential for achieving the best rigidity results.

8.1.3 Characterizing robustness of rigidity analysis results.

A rigid cluster which is *robust* is unlikely to break apart with the loss of any interaction. The prevalence such 'critical' interactions in the rigidity results of real protein data was previously unknown.

We proposed a new method, which we call *redundancy analysis*, which measures the ratios of redundant and non-redundant interactions. First the method classifies each interaction in a cluster as *critical* or *redundant*. An interaction is critical if its removal causes the cluster to lose rigidity, breaking apart into smaller clusters. If its removal has no effect of the cluster's rigidity, the interaction is classified as redundant. Then the interaction uses the counts to compute a *redundancy score* for each cluster.

We performed redundancy analysis, with KINARI v1.0 parameter settings, on the largest rigid clusters of the Pdomain benchmark data set. The evaluation confirmed our intuitions that, using these parameter settings, the small-sized clusters had lower redundancy than the medium- and large-sized clusters. In this analysis, we also keep track of the change in cluster size upon an interaction's removal, which we called the interaction's *criticality value*. Clusters containing interactions with high criticality values ($> 10\%$) were rare, and when these did occur, it was worth a closer inspection. We examined the PDBs in the Gerstein Lab data set in which the largest rigid clusters had higher criticality values. We found that these interactions were typically clustered together around the active site. They show that a delicate balance between rigidity

and flexibility is maintained in these sites, as opposed to other rigid regions in the protein.

BIBLIOGRAPHY

[1] Alexej Abyzov, Robert Bjornson, Mihali Felipe, and Mark Gerstein. RigidFinder: a fast and sensitive method to detect rigid blocks in large macromolecular complexes. Proteins: Structure, Function, and Bioinformatics, 78(2):309–324, February 2010.

[2] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, pages 79–85, 1998.

[3] William A. Beard and Samuel H. Wilson. Structure and mechanism of DNA Polymerase Beta. Chemical Reviews, 106(2):361–382, 2006. PMID: 16464010.

[4] Carl Branden and John Tooze. Introduction to Protein Structure. Garland Publishing, Inc., New York, second edition, 1998.

[5] Bernard Brooks and *et al.* CHARMM: The Biomolecular Simulation Program. Journal of Computational Chemistry, 30(10, Sp. Iss. SI):1545–1614, 2009.

[6] I. David Brown and Brian McMahon. CIF: the computer language of crystallography. Acta Crystallographica Section B: Structural Science, 58(3 Part 1):317–324, Jun 2002.

[7] Sabrina Bédard, Leland C. Mayne, Ronald W. Peterson, A. Joshua Wand, and S. Walter Englander. The foldon substructure of staphylococcal nuclease. Journal of Molecular Biology, 376(4):1142 – 1154, 2008.

[8] David A. Case and *etal*. The Amber biomolecular simulation programs. Journal of Computational Chemistry, 26(16):1668–1688, December 2005.

[9] Vincent B. Chen, W. Bryan Arendall, Jeffrey J. Headd, Daniel A. Keedy, Robert M. Immormino, Gary J. Kapral, Laura W. Murray, and David C. Richardson. MolProbity: all-atom structure validation for macromolecular crystallography. Acta Crystallographica. Section D, Biological Crystallography, 66:12–21, 2010.

[10] M. V. Chubynsky, B. M Hespenheide, Donald J. Jacobs, Leslie A. Kuhn, Ming Lei, Scott Menor, A. J. Rader, Michael F. Thorpe, Walter Whiteley, and Maria I. Zavodszky. Constraint theory applied to proteins. Nanotechnology Research Journal, 2:61–72, 2008.

[11] M. V. Chubynsky and Michael F. Thorpe. Algorithms for 3d rigidity analysis and a first order percolation transition. Physical Review E, 76(041135), 2007.

[12] Pamela Clark, Jessica Grant, Samantha Monastra, Filip Jagodzinski, and Ileana Streinu. Periodic rigidity of protein crystal structures. In 2nd IEEE International Conference on Computational Advances in Bio and Medical Sciences (ICCABS'12). Feb. 23-25, February 2012.

[13] Qiang Cui and Ivet Bahar, editors. Normal mode analysis : theory and applications to biological and chemical systems. Chapman & Hall/CRC, 2006.

[14] Aleksandar Cvetkovic, Angeli Lal Menon, Michael P. Thorgersen, Joseph W. Scott, Farris L. Poole II, Francis E. Jenney Jr, W. Andrew Lancaster, Jeremy L. Praissman, Saratchandra Shanmukh, Brian J. Vaccaro, Sunia A. Trauger, Ewa Kalisiak, Junefredo V. Apon, Gary Siuzdak, Steven M. Yannone, John A. Tainer, and Michael W. W. Adams. Microbial metalloproteomes are largely uncharacterized. Nature, 466(7307):779–782, August 2010.

[15] Carlos A. Del Carpio, A. R. Shaikh, Eiichiro Ichiishi, M. Koyama, K. Nishijima, and Akira Miyamoto. A graph theoretical approach for analysis of protein flexibility change at protein complex formation. Genome Informatics, 16(2):148–160, 2005.

[16] Andreas W. M. Dress, A. Dreiding, and H. Haegi. Classification of mobile molecules by category theory. Studies in Physical and Theoretical Chemistry, 23:39–58, 1983.

[17] Michel Dumontier, Rong Yao, Howard J. Feldman, and Christopher W.V. Hogue. Armadillo: Domain boundary prediction by amino acid composition. Journal of Molecular Biology, 350(5):1061 – 1073, 2005.

[18] Samuel Flores, Duncan Millburn, Nathaniel Echols, B. M Hespenheide, Kevin Keating, Jason Lu, Stephen A. Wells, Eric Z. Yu, Michael F. Thorpe, and Mark Gerstein. The database of macromolecular motions: new features added at the decade mark. Nucleic Acids Research, 34:D296–D301, 2006.

[19] Samuel C Flores and Mark B Gerstein. FlexOracle: predicting flexible hinges by identification of stable domains. BMC Bioinformatics, 8, 2007.

[20] Samuel C Flores, Long J. Lu, Julie Yang, Nicholas Carriero, and Mark B Gerstein. Hinge Atlas: relating protein sequence to sites of structural flexibility. BMC Bioinformatics, 8, 2007.

[21] Naomi Fox, Filip Jagodzinski, Yang Li, and Ileana Streinu. KINARI-Web: A Server for Protein Rigidity Analysis. Nucleic Acids Research, 39(Web Server Issue), 2011.

[22] Naomi Fox, Filip Jagodzinski, and Ileana Streinu. KINARI-Lib: a C++ library for pebble game rigidity analysis of mechanical models. In Minisymposium on Publicly Available Geometric/Topological Software, Chapel Hill, NC, USA, Jun. 17-19, 2012.

[23] Naomi Fox and Ileana Streinu. Redundant interactions in protein rigid cluster analysis. In 1st IEEE International Conference on Computational Advances in Bio and medical Sciences (ICCABS). Feb. 3-5, 2011, February 2011. DOI 10.1109/ICCABS.2011.5729952.

[24] Naomi Fox and Ileana Streinu. Towards accurate modeling for protein rigidity analysis. In 2nd IEEE International Conference on Computational Advances in Bio and Medical Sciences (ICCABS'12). Feb. 23-25, February 2012.

[25] Deborah S. Franzblau. Combinatorial algorithm for a lower bound on frame rigidity. SIAM Journal on Discrete Mathematics, 8(3):388–400, 1995.

[26] Deborah S. Franzblau. Generic rigidity of molecular graphs via ear decomposition. Discrete Applied Mathematics, 101(1-3):131–155, 2000.

[27] Darón I. Freedberg, Rieko Ishima, Jaison Jacob, Yun-Xing Wang, Irina Kustanovich, John M. Louis, and Dennis A. Torchia. Rapid structural fluctuations of the free HIV protease flaps in solution: Relationship to crystal structures and comparison with predictions of dynamics calculations. Protein Science, 11(2):221–232, 2002.

[28] Daan Frenkel and B. Smit. Understanding Molecular Simulation (Computational Science Series, Vol 1). Academic Press, October 2001.

[29] Simone Fulle and Holger Gohlke. Analyzing the flexibility of rna structures by constraint counting. Biophysics Journal, 94:4202–4219, 2008.

[30] Harold N. Gabow and Herbert Hans Westermann. Forests, frames, and games: algorithms for matroid sums and applications. In Proceedings of the twentieth annual ACM symposium on theory of computing (STOC'88), pages 407–421. ACM Press, 1988. http://dx.doi.org/10.1145/62212.62252.

[31] Gastone Gilli and Paola Gilli. The Nature of the Hydrogen Bond, Outline of a Comprehensive Hydrogen Bond Theory. Oxford University Press, 2009.

[32] Holger Gohlke, Leslie A. Kuhn, and David A. Case. Change in protein flexibility upon complex formation: analysis of Ras-Raf using molecular dynamics and a molecular framework approach. Proteins, 56(2):322–337, August 2004.

[33] Holger Gohlke and S. Radestock. Exploiting the link between protein rigidity and thermostability for data-driven protein engineering. Engineering in Life Sciences, 8:507–522, 2008.

[34] Holger Gohlke and Michael F. Thorpe. A natural coarse graining for simulating large biomolecular motion. Biophysical Journal, 91(6):2115–2120, 2006.

[35] Luis C. Gonzalez, Hui Wang, D. R. Livesay, and Donald J. Jacobs. Calculating ensemble averaged descriptions of protein rigidity without sampling. PLoS One, 7(2), 2012.

[36] Kannan Gunasekaran, Arnold T. Hagler, and Lila M. Gierasch. Sequence and structural analysis of cellular retinoic acid-binding proteins reveals a network of conserved hydrophobic interactions. Proteins: Structure, Function, and Bioinformatics, 54(2):179–194, 2004.

[37] Kirk Haller, Audrey Lee-St. John, Meera Sitharam, Ileana Streinu, and Neil White. Body-and-cad geometric constraint systems. Computational Geometry: Theory and Applications, accepted, 2010.

[38] Bruce Hendrickson. The molecule problem: determining conformation from pairwise distances. PhD thesis, Cornell University, 1991.

[39] Katherine Henzler-Wildman and Dorothee Kern. Dynamic personalities of proteins. Nature, 450(7172):964–72, 2007.

[40] B. M Hespenheide, A. J. Rader, Michael F. Thorpe, and Leslie A. Kuhn. Identifying protein folding cores: Observing the evolution of rigid and flexible regions during unfolding. Journal of Molecular Graphics and Modelling, 21(3):195–20, 2002.

[41] Berk Hess, Carsten Kutzner, D. van der Spoel, and Erik Lindahl. Gromacs 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. Journal of Chemical Theory and Computation, 4(3):435–447, 2008.

[42] Timothy A. Holland, Stella Veretnik, Ilya N. Shindyalov, and Philip E. Bourne. Partitioning protein structures into domains: Why is it so difficult? Journal of Molecular Biology, 361(3):562–590, 2006.

[43] Viktor Hornak, Asim Okur, Robert C. Rizzo, and Carlos Simmerling. HIV-1 protease flaps spontaneously open and reclose in molecular dynamics simulations. Proceedings of the National Academy of Sciences, 103(4):915–920, 2006.

[44] Hiroshi Imai. On combinatorial structures of line drawings of polyhedra. Discrete Applied Mathematics, 10:79–92, 1985.

[45] Donald J. Jacobs. Generic rigidity in three-dimensional bond-bending networks. Journal of Physics A: Mathematical and General, 31:6653–6668, 1998.

[46] Donald J. Jacobs and Bruce Hendrickson. An algorithm for two-dimensional rigidity percolation: the pebble game. Journal of Computational Physics, 137:346–365, November 1997.

[47] Donald J. Jacobs, Leslie A. Kuhn, and Michael F. Thorpe. Flexible and rigid regions in proteins. In Rigidity Theory and Applications, pages 357–384. Kluwer Academic, 1999.

[48] Donald J. Jacobs, A. J. Rader, Michael F. Thorpe, and Leslie A. Kuhn. Protein flexibility predictions using graph theory. Proteins, 44:150–165, 2001.

[49] Donald J. Jacobs and Michael F. Thorpe. Generic rigidity percolation: The pebble game. Physical Review Letters, 75(22):4051–4054, 1995.

[50] Donald J. Jacobs and Michael F. Thorpe. Computer-implemented system for analyzing rigidity of substructures within a macromolecule. United States Patent 6014449, January 2000.

[51] Filip Jagodzinski, Jeanne Hardy, and Ileana Streinu. Using rigidity analysis to probe mutation-induced structural changes in proteins. Journal of Bioinformatics and Computational Biology, 10(3), 2012.

[52] George Jeffrey and Wolfram Saenger. Hydrogen Bonding in Biological Structures. Springer-Verlag, 1991.

[53] P. A. Karplus and G. E. Schulz. Prediction of chain flexibility in proteins. Naturwissenschaften, 72:212–213, 1985.

[54] Naoki Katoh and Shin ichi Tanigawa. A proof of the molecular conjecture. In Symposium on Computational Geometry, pages 296–305, 2009.

[55] Naoki Katoh and Shinichi Tanigawa. A proof of the molecular conjecture. Discrete and Computational Geometry, 45(4):647–700, 2011.

[56] Kevin Keating, Samuel Flores, Mark Gerstein, and Leslie A. Kuhn. Stonehinge: Hinge prediction by network analysis of individual protein structures. Protein Science, 18(2):359–371, 2008.

[57] John L Klepeis, Kresten Lindorff-Larsen, Ron O Dror, and David E Shaw. Long-timescale molecular dynamics simulations of protein structure and function. Current Opinion in Structural Biology, 19(2):120 – 127, 2009.

[58] Tanja Kortemme, Alexandre V. Morozov, and David Baker. An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein-protein complexes. Journal of Molecular Biology, 326(4):1239–1259, February 2003.

[59] Julio A. Kovacs, Pablo Chacon, and Ruben Abagyan. Predictions of protein flexibility: first-order measures. Proteins: Structure, Function, and Bioinformatics, 56(1):661–668, September 2004.

[60] Dennis M. Kruger, Aqeel Ahmed, and Holger Gohlke. NMSim web server: integrated approach for normal mode-based geometric simulations of biologically relevant conformational transitions in proteins. Nucleic Acids Research, 40 (Web Server issue):W310–W316.

[61] Maria Kurnikova, Tatyana Mamanova, B. M Hespenheide, and Rachel Straub. Protein flexibility using constraints from molecular dynamics simulations. Physical Biology, 2(4):S137–S147, 2005.

[62] Gerard Laman. On graphs and rigidity of plane skeletal structures. Journal of Engineering Mathematics, 4:331–340, 1970.

[63] Sol Lederer, Yue Wang, and Jie Gao. Connectivity-based localization of large scale sensor networks with complex shape. In Proceedings of the 27th Annual IEEE Conference on Computer Communications (INFOCOM'08), pages 789–797, May 2008.

[64] Audrey Lee and Ileana Streinu. Pebble game algorithms and sparse graphs. Discrete Mathematics, 308(8):1425–1437, April 2008.

[65] Audrey Lee, Ileana Streinu, and Louis Theran. Analyzing rigidity with pebble games. In Symposium on Computational Geometry, pages 226–227, New York, NY, USA, 2008. ACM.

[66] R. A. Lee, M. Razaz, and S. Hayward. The DynDom database of protein domain motions. Bioinformatics, 19(10):1290–1291, July 2003.

[67] Haripada Maity, Mita Maity, Mallela M. G. Krishna, Leland Mayne, and S. Walter Englander. Protein folding: The stepwise assembly of foldon units. Proceedings of the National Academy of Sciences, 102(13):4741–4746, March 2005.

[68] S. L. Mayo, B. I. Dahiyat, and D. B. Gordon. Automated design of the surface positions of protein helices. Protein Science, 6(6):1333–1337, 1997.

[69] IK McDonald and Janet M. Thornton. Satisfying hydrogen bonding potential in proteins. Journal of Molecular Biology, 238(5):777–793, May 1994.

[70] Cristian F. Moukarzel. An efficient algorithm for testing the generic rigidity of graphs in the plane. Journal of Physics A: Mathematical and General, 29:8079–8098, December 1996.

[71] A Mozzarelli and G L Rossi. Protein function in the crystal. Annual Review of Biophysics and Biomolecular Structure, 25(1):343–365, 1996. PMID: 8800474.

[72] Haruki Nakamura. Roles of electrostatic interaction in proteins. Quarterly Reviews of Biophysics, 29:1–90, 1996.

[73] SK Panigrahi and GR Desiraju. Strong and weak hydrogen bonds in the protein-ligand interface. Proteins: Structure, Function, and Bioinformatics, 67(1):128–141, 2007.

[74] Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description Version 3.20. online, October 2008.

[75] Gregory A Petsko and Dagmar Ringe. Protein Structure and Function. New Science Press, 2004.

[76] A. J. Rader, Gulsum Anderson, Basak Isin, Ivet Bahar, and Judith Klein-Seetharaman. Identification of core amino acids stabilizing rhodopsin. Proceedings of the National Academy of Sciences, 101(19):7246–7251, May 2004.

[77] A. J. Rader and Ivet Bahar. Folding core predictions from network models of proteins. Polymer, 45(2):659–668, January 2004.

[78] A. J. Rader, B. M Hespenheide, Leslie A. Kuhn, and Michael F. Thorpe. Protein unfolding: Rigidity lost. Proceedings of the National Academy of Sciences, 99:3540–3545, 2002.

[79] Sebastian Radestock and Holger Gohlke. Protein rigidity and thermophilic adaptation. Proteins: Structure, Function, and Bioinformatics, 79(4):1089–1108, 2011.

[80] Ileana Streinu, Filip Jagodzinski, and Naomi Fox. Tutorial: Analyzing protein flexibility: an introduction to combinatorial rigidity methods and applications. In IEEE International Conference Bioinformatics & Biomedicine (BIBM 2011), Atlanta, GA, Nov 12-15, 2011, 2011.

[81] Kokichi Sugihara. On redundant bracing in plane skeletal structures. Bull. Electrotech. Lab., 44:376–386, 1980.

[82] John A. Tainer, Victoria A. Roberts, and Elizabeth D. Getzoff. Protein metal-binding sites. Current Opinion in Biotechnology, 3(4):378–387, 1992.

[83] Tiong-Seng Tay. Review: Rigidity problems in bar and joint frameworks and linkages of rigid bodies. Structural Topology, 8:33–36, 1983.

[84] Tiong-Seng Tay. Rigidity of multigraphs I: linking rigid bodies in n-space. Journal of Combinatorial Theory, Series B, 26:95–112, 1984.

[85] Tiong-Seng Tay. Linking $(n-2)$-dimensional panels in n-space II: $(n-2,2)$-frameworks and body and hinge structures. Graphs and Combinatorics, 5:245–273, 1989.

[86] Tiong-Seng Tay and Walter Whiteley. Recent advances in the generic rigidity of structures. Structural Topology, 9:31–38, 1984.

[87] Shawna Thomas, Xinyu Tang, Lydia Tapia, and Nancy M. Amato. Simulating protein motions with rigidity analysis. Journal Of Computational Biology, 14(6):839–855, 2007.

[88] FIRST 6.2.1 User Guide. `http://flexweb.asu.edu/software/first/user_guides/FIRST6.2.1_user_guide.pdf`, 2009.

[89] Michael F. Thorpe, M. V. Chubynsky, B. M Hespenheide, Scott Menor, Donald J. Jacobs, Leslie A. Kuhn, Maria I. Zavodszky, Ming Lei, A. J. Rader, and Walter Whiteley. Flexibility in Biomolecules, chapter 6, pages 97–112. Current Topics in Physics. Imperial College Press, London, 2005. R.A. Barrio and K.K. Kaski, eds.

[90] Michael F. Thorpe, B. M Hespenheide, Yi Yang, and Leslie A. Kuhn. Flexibility and critical hydrogen bonds in cytochrome c. Pacific Symposium on Biocomputing., pages 191–202, 2000.

[91] Mauno Vihinen, Esa Torkkila, and Pentti Riikonen. Accuracy of protein flexibility predictions. Proteins, 19(2):141–149, 1994.

[92] Vincent A Voelz, Gregory R. Bowman, Kyle Beauchamp, and Vijay S. Pande. Molecular simulation of ab initio protein folding for a millisecond folder ntl9(1-39). Journal of the American Chemical Society, 132(5):1526–1528, 2010. PMID: 20070076.

[93] Irene T. Weber and J. Agniswamy. HIV-1 Protease: Structural Perspectives on Drug Resistance. Viruses, 1(3):1110–1136, December 2009.

[94] Stephen A. Wells, J. E. Jimenez-Roldan, and R A Romer. Comparative analysis of rigidity across protein families. Physical Biology, 6(4):046005, 2009.

[95] Stephen A. Wells, Scott Menor, B. M Hespenheide, and Michael F. Thorpe. Constrained geometric simulation of diffusive motion in proteins. Physical Biology, 2:S127–S136, 2005.

[96] J. Michael Word, Jane S. Richardson, David C. Richardson, and Simon C. Lovell. Asparagine and glutamine: using hydrogen atom contacts in the choice of sidechain amide orientation. Journal of Molecular Biology, 285:1735–1747, 1999.

[97] Willy Wriggers and Klaus Schulten. Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. Proteins: Structure, Function, and Bioinformatics, 29(1):1–14, December 1998.

[98] Semen O. Yesylevskyy and Valery N. Kharkyanen. Fuzzy domains: New way of describing flexibility and interdependence of the protein domains. Proteins: Structure, Function, and Bioinformatics, 74(4):980–995, 2009.